AD-A118 168 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 20/4
APPLICATION OF A DISCRETE NONLINEAR SPECTRAL MODEL TO IDEAL CAS--ETC(U)
APR 82 J H ALLENDER, M LYBANON
UNCLASSIFIED NORDA-TN-148 NL

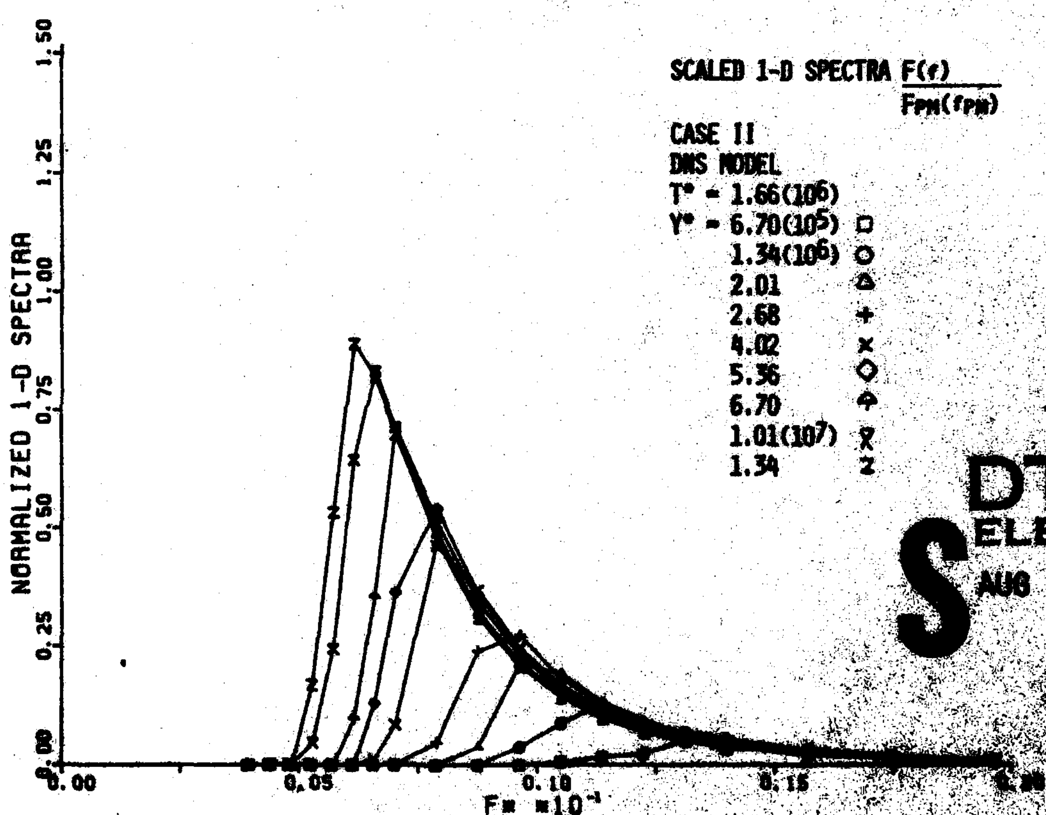
1 OF 3
AD
A118168

AD A118168

[illegible]

Naval Ocean Research and
Development Activity
NSTL Station, Mississippi 39529

Application of a Discrete Nonlinear Spectral Model to Ideal Cases of Wind Wave Generation



DTIC FILE COPY

J.H. Allender [illegible]

Oceanography Division
Ocean Science and Technology Laboratory

April 1982

82 08 13 [illegible]

ABSTRACT

The discrete nonlinear spectral model developed by Allender, Barnett, and Lybanon ~~(Proc~~. Symp. on Wave Dyn., Plenum Press, 1982) is applied to seven ideal cases of wind wave generation. These cases form the basis of an international study to compare wave prediction models, ~~(Hasselmann~~ et al., Proc. Symp. on Wave Dyn., Plenum Press, 1982). Model results are summarized graphically for each case, as applicable, and a brief explanation of model behavior is given.

The model is free of numerical dispersion and damping, and reproduces the JONSWAP relations for fetch- and duration-limited wave growth quite well. Model results from cases with increasing complexity provide a benchmark for understanding model behavior in actual situations and for future improvements.

ACKNOWLEDGEMENTS

T.P. Barnett is thanked heartily for being an enthusiastic and optimistic collaborator throughout this project. Isaac Traxler did all of the computer graphics. Michael Stirgus and Richard Myrick put the plots into final form. Joyce Ford is thanked for typing the manuscript.

CONTENTS

A SUMMARY OF CALCOMP-COMPATIBLE PLOT PROGRAMS FOR THE WAVE MODEL

Isaac Traxler

Computer Sciences Corporation

NSTL Station, MS 39529

LIST OF ILLUSTRATIONS

LIST OF TABLES

APPLICATION OF A DISCRETE NONLINEAR SPECTRAL MODEL TO IDEAL CASES OF WIND WAVE GENERATION

I. INTRODUCTION

A variety of wave prediction models have been developed in recent years. These models differ considerably in the way they represent various physical processes and also in their manner of dealing with all the messy details inherent in realistic wave prediction.

In order to assess model performance and identify critical areas for future research efforts, a joint investigation among wave modelers was undertaken. Nine groups of modelers applied their models to seven ideal cases of wind wave generation and then conducted a detailed intercomparison of model results. A preliminary report of the model intercomparison was given at the Symposium on Wave Dynamics and Radio Probing of the Ocean Surface, Miami, FL, May, 1981. The final report appears in the proceedings of that symposium (Hasselmann, et al., 1982).

NORDA Code 331 was one of the participants in the intercomparison. We applied the discrete nonlinear spectral (DNS) model, developed by Code 331 in collaboration with T.P. Barnett of Scripps Institute of Oceanography, to the ideal cases. The model physics and underlying philosophy is described in Allender et al. (1982). A detailed description of the model computer program is given in NORDA Technical Note 147.

For the present model we assume that input from the atmosphere S_{in} and weak, nonlinear interactions among waves S_{nl} are the dominant source terms that govern the growth of the wave spectrum. Dissipation, i.e., any process that removes energy from the surface wave field, enters only through a prescribed, maximum spectral

density that depends on the local wind speed. The form of S_{in} yields an exponential growth mechanism, consistent with linear feedback theories. The magnitude of S_{in} is consistent with the minimum suggested by Snyder et al. (1981). The nonlinear interactions are represented by a set of empirical orthogonal functions that decompose a body of exact calculations that were made for a family of single-peaked spectra with different peakedness and angular spread. Thus, the representation for S_{nl} depends crudely on spectral shape. The Pierson-Moskowitz spectrum with its peak at the local wind frequency, but with a variable energy level and a $\cos^2$ angular spread with respect to the wind direction, is used as the spectral limiter. Some additional constraints (on the rate of change of the spectrum and on where in frequency and direction to apply the limiter) are also required to obtain stable spectra. These constraints are borne out of the mismatch between the relatively large amount of information that ought to be present in the 2-D wave spectrum and the relatively low level of sophistication used to represent physical processes affecting wave growth.

The algorithm used for propagation is generally the most complicated aspect of numerical methods in wave models. We use an adaptation of the method of characteristics to avoid the numerical damping and dispersion problems that are typical of finite difference methods. The wave field is quantized as sets of frequency-direction particles that move along parallel rays in given directions and at the group velocities of the given frequency components. The rays are thought of as directed line segments that connect equally spaced ray points. A net of grid points is also defined over the physical domain to introduce the wind field, to accumulate values of the 2-D spectrum and to compute the source terms. Reciprocal relationships between nearest neighbor ray and grid points then form the backbone of the overall algorithm.

The primary purpose of the present note is to describe briefly and to present all pertinent DNS model results obtained for the ideal cases. The note stands as a record of model performance, and can be used to better understand the effect of modifications to the DNS model and the results of future model applications. The format for the remainder of the note is a case by case presentation of results.

II. PRESENTATION OF MODEL RESULTS

A. CASE I, PROPAGATION

The DNS model uses an adaptation of the method of characteristics to propagate spectral density (which is proportional to surface displacement squared per elemental frequency and direction increments). The purpose of CASE I was to reveal numerical damping and dispersion inherent in the numerical algorithm, which by definition should be no larger than roundoff error in the present model.

We tested the model, however, by comparing results with an analytical solution. If we assume that spectral density F propagates in direction x with (group) velocity C_g and can grow in an exponential fashion controlled by a constant, say B, then

$$\frac{\partial F}{\partial t} + C_g \frac{\partial F}{\partial x} = BF \tag{1}$$

subject to initial and boundary conditions on $0 \leq x \leq L$. For simplicity we take $F(0,t) = K$ and $F(x,0) = g(x)$.

Then the solution to (1) is

$$F(x,t) = \begin{cases} \exp(Bt)\, g(x - C_g t) & x \geq C_g t \quad (2) \\ K \exp(Bx/C_g) & x < C_g t \quad (3) \end{cases}$$

Specifically, in the interval $0 \leq x \leq L$:

1. If $t \leq L/C_g$ there are two regimes

   $F(x,t)$ propagates the initial condition and grows it exponentially in time for $x \geq C_g t$, $F(x,t) = e^{Bt} g(x - C_g t)$;

   $F(x,t)$ is constant in time but increases exponentially with distance for $x < C_g t$, and

2. If $t > L/C_g$ then "information" from $x = 0$ has had time to reach all points in $[0, L]$, and $F(x,t) = Ke^{Bx/C_g}$ everywhere.

We specified a constant boundary condition for several frequencies and directions and ran the model, suppressing all source functions for spectral density except for atmospheric input (which is proportional to BF). At first the model solution did not agree with Eqs. (2)-(3). An error in the linkage arrangement between ray and grid points was discovered and corrected (see Allender and Lybanon, 1982, Secs. II.A, IV.A, and V.A for details on ray-grid concept). The corrected model satisfied Eqs. (2)-(3) to within round-off error in our tests and we concluded that the model propagates energy at the specified group velocities without significant numerical damping or dispersion.

B. FETCH-LIMITED GROWTH

The results of comprehensive studies of wave growth under fetch-limited conditions (Hasselmann et al., 1976) provide benchmarks for evaluating the performance of wave prediction models. In this case we have appriori knowledge about the relationships among variance, peak, fetch, and time, and, to a lesser extent, some quasi-equilibrium form for the growing wind sea spectrum (Hasselmann et al., 1976).

The physical setting for CASE II was a 1000 x 1000 km square ocean, initially at rest, subjected to a constant offshore wind of 20 m/s. The upwind boundary was treated as a shoreline (solid), whereas the other three boundaries were perfect

absorbers. Table 1 gives the frequencies, directions, and other model parameters that were used. Fig. 1 gives a schematic of the domain and the locations selected for model output.

Model results for CASE II are summarized in Figs. 2-15. The figures are largely self-explanatory and a table of mathematical notation is included as Sec. V.

TABLE 1

MODEL PARAMETERS FOR CASES II-V

Domain	1000 by 1000 km; no flux condition on S boundary with perfect absorption for outgoing rays on W, N, E boundaries, initially at rest.
Grid	50 km, regular
Rays	30 parallel rays, ea. dir. 35-40 km ray-point spacing
Time Step	1200 sec
Run Time	40 hr
Frequencies (20)	.045, .05, ..., .08, .09, ..., .16, .18, .225, .25 hz
Directions (24)[1]	0, 15, 30, ..., 330, 345°

[1]A resolution of 45° was used in the half plane opposite the wind direction in CASES II-IV and in the SE quadrant in CASE V.

Key results for this simulation are given in Figs. 2-3, which show variance E* and peak frequency f_p* vs. fetch x* (all variables nondimensional) for successive times during the growth of the spectrum. Good agreement with the JONSWAP relations (plotted in the figures) is found over most of the fetch. Such agreement is a significant improvement over previous spectral models. Notably, the model reproduces the JONSWAP relations by combining processes that depend on five independent variables (f,θ,x,y,t) and <u>without</u> tuning of the model parameters. Model spectra are still fetch-limited at $x^*=1.34(10^7)$. A fetch of about $x^*=1.6(10^7)$ would be required for full development. Figs. 4 and 6 show the behavior of E* and f_p vs. t*. As seen in Figs. 2, 3, 4 and 6, the model E* and f_p* values scatter somewhat for fetches less than about two grid lengths because the gradients in the wave field are not resolved. The situation is similar to that occurring in discrete hydrodynamical modeling, but not as severe because some wave components actually can grow to their limit in just a few temporal or spatial steps. To reduce the spatial truncation errors we ran a short test on a 200 x 200 km domain with a 10 km grid and a 10 min time step, and all other model parameters as before. With this five-fold increase in resolution there is no scatter in E* or f_p* at short fetches as shown in Figs. 5 and 7, respectively. In practical cases the difficulty could be corrected altogether by simply setting the high frequencies to their limiting values, as described in Allender et al., 1982, Sec. 3.

The times at which the wave field becomes fetch-limited agree well with a simple relation that can be derived from the JONSWAP duration law reported by Hasselmann et al. (1976). To use this relation we assume that the fetch-limited domain expands at the group velocity of the peak frequency. Then the extent of the fetch-limited domain X_{FL} as a function of time, starting with no wind sea, is found to be:

$$X_{FL} = 0.0021\ U_*^2\ t^{*(10/7)} \qquad (4)$$

using a constant drag coefficient of $1.8(10^{-3})$ to convert to U_*. The times at which E^* and f_p^* become time-dependent in Figs. 2-3 are predicted well by Eqn. (4) for $x^*>1.34(10^6)$. This prediction accurately implies that the present model also follows the JONSWAP duration laws.

Figs. 8-9 provide a convenient summary of the development of variance and peak, normalized by the 'final' values defined according to the Pierson-Moskowitz spectrum, vs. both fetch and time. A smooth progression of E^* and f_p^* values is seen except for short fetches as explained in the preceeding paragraphs.

The development of 1-D model spectra (energy density vs. frequency) is shown in Figs. 10-11. The peak enhancement factor (cf. Hasselmann et al., 1976) of one-dimensional model spectra decreases more or less monotonically from about 2.7 to 1.5 between $x^*=1.34(10^6)$ and $x^*=1.34(10^7)$. Wave components greater than about 1.3-1.4 f_p^* tend to reach their limiting values as the model spectra develop. So, the high frequency side of the spectrum follows an f^{-5} slope except in a narrow band to the right of the peak. In this band densities are somewhat higher than for corresponding JONSWAP spectra, perhaps because S_{net} is not allowed to be negative here.

Examples of 2-D model spectra are given in Figs. 12-15. Some of the raggedness typical of short fetches shows up as well as the perfect symmetry imposed by the limiting form of the spectrum at long fetches.

C. CASE III, OBLIQUE FETCH-LIMITED GROWTH

The physical domain and model configuration were the same as for CASE II (1000 x 1000 km square ocean, model parameters as in Table 1), except that the wind blew at an angle of 45° seaward off the straight coastline. A sketch of the domain

and output points is given in Fig. 16. The primary item of interest was whether the model predicted skew spectra at short fetches (perpendicular to the shoreline). A secondary interest was growth with distance as measured in the wind direction (toward NE) starting from the SE corner of the domain. Presumably this growth follows the same fetch law as in CASE II.

All of the model results prepared for the model intercomparison are presented in Figs. 17-33.

Growth with distance downwind (Fig. 27) compares well with growth for the corresponding fetches from CASE II (Fig. 10). The raggedness in 2-D spectra caused by irregularities in f_p^* at short fetches is evident in Fig. 28, for example. The model does indeed predict skew spectra for areas close to the shoreline with long <u>alongshore</u> fetches. Fig. 30 demonstrates this skewness with higher frequencies aligned with the wind direction (45°) and lower frequencies running nearly parallel to the shoreline (90°).

D. CASE IV, WINDY HALF-PLANE

The model domain was a 1000 x 1000 km square ocean, initially at rest, subject to a constant wind of 20 m/s in the left (western) half plane, and no wind in the right (eastern) half plane (See Table 1 for model parameters). Fig. 34 gives a sketch to locate output points. The overriding purpose of this case was to examine, qualitatively, at least the propagation of swell into the half plane with no wind.

All of the model results extracted for the intercomparison are shown in Figs. 35-49. A visual impression of growth of variance and the average angle of the spectrum appears in the so-called 'General Custer' plot, Fig. 37. (Participants in the international, wave model study seemed quite willing to carry along this dubious

name, although some of them were not familiar with the historical background.) The model response is symmetric about the N-S centerline of the windy half plane. Swells propagate into the calm half plane, their energetic constituents being more aligned with the wind direction as fetch increases. The ragged 2-D spectrum at short fetches is again exemplified in Fig. 38. Some locations show two almost distinct swell spectra, e.g., Fig. 45, a somewhat unrealistic situation that is modified by geometric dispersion in actuality.

Detailed behavior in the eastern half plane seems to depend primarily on two things: 1) the basic growth curve followed by spectral components under constant wind, and 2) the assumed angular distribution for the limiting spectrum, especially as it affects larger fetches. The behavior also depends on the strength of coupling among different wave directions through our representation for nonlinear transfer.

E. CASE V, FRONTAL PASSAGE

The model domain was a 1000 x 1000 km square ocean, initially at rest, subject to a constant S wind of 20 m/s below the SW-NE bisector of the domain and an E wind of 20 m/s above the bisector. (See Table 1 for model parameters.) Fig. 50 sketches the situation and locates the selected output points. This case was designed primarily to test the manner in which the model wave field responded to a change in wind direction, now known as 'directional relaxation'.

Figs. 51-63 were prepared for the intercomparison. The overall response can be seen quickly in Fig. 52, the General Custer diagram. The variance grows with increasing fetch toward the north until the front is crossed. The variance drops, then the average wave direction relaxes to the new direction, as the variance grows again. The relaxation is faster for less developed spectra, y^* about $0.25(10^7)$, than for more fully-developed spectra, y^* about $0.75(10^7)$.

In retrospect we think that the drop in variance upon crossing the front is unrealistic, although the average angle response hit the middle ground among all the different wave model results (Hasselmann et al., 1982). Directional relaxation in the model is affected by the nonlinear transfer term in the model equations, which is applied with respect to the average wave direction (not wind direction). Also, a variety of logic statements, used in applying the limiting spectrum affect the relaxation process through imposed (and sometimes) abrupt changes in spectral density at various f and θ. Some rethinking of the model is necessary here, as well as comparison of model behavior with limited theory and observations, such as they are.

In this very complex case bimodal seas arise in the SW part of the domain above the front, e.g., Fig. 56. This behavior was seen in most of the models tested in the intercomparison. Some isolated spurious peaks were noted, especially in the NW part of the domain, e.g., Fig. 60. These peaks were apparently caused by temporary changes to the model (Allender and Lybanon, 1982), made in lieu of using some mechanism for subgrid scale wave growth. Looking back, it seems likely that subgrid growth would have contributed to a more reasonable overall response to this case.

F. CASE VI A, STATIONARY HURRICANE

Two model hurricanes were studied. The (synthetic) wind field (prepared by V. Cardone, Oceanweather, Inc., White Plains, N.Y.) was the same for both cases except that the storm was stationary in CASE VI A and translated at 15 m/s to the north in CASE VI B. The spatial distribution of wind speed and direction was patterned after a northward-moving hurricane, with a maximum speed of 40 m/s located 50 km northeast of center, a pronounced east-west wind-speed asymmetry (stronger east of center) and a pronounced north-south wind-direction asymmetry (more inflow in rear quadrants).

The model domain was defined as $270 \leq x \leq 1030$ km east-west by $700 \leq y \leq 1700$ km north-south with a grid spacing of 40 km (selected to resolve the wind field). Table 2 gives other pertinent model parameters. The stationary storm was located at (650, 1400) for a duration of 24 hr. The parametric hurricane model by Ross (1976) was used for initial and boundary conditions throughout the run. The Ross model defines the total energy and peak frequency at any point in a hurricane in terms of the wind speed and radial distance from the eye. The directional spectrum in that model is taken to be of JONSWAP type with a $\cos^2$ spread relative to the local wind direction. In addition we set the spectrum fully-developed initially at the eye in this case. Clearly, the Ross method colored our results, because of the relatively small model domain that was chosen out of necessity (maximum number of grid points without recoding). We are encouraged, however, by the reasonable results that we obtained in both cases.

Figure 64 shows the relative locations of selected output points. Figs. 65-84 were prepared for the model intercomparison. Significant wave heights between 9 and 10 m (Fig. 65) and average frequencies of about 0.1 hz (Fig. 66) occur roughly northeast of the eye of the storm. The high average frequencies along the southern boundary are artifacts produced by our treatment of the source terms. If the wind speed is less than 5 knots, no atmospheric input is added. If no low or mid frequency bands contain significant energy the nonlinear transfer will pump energy from the spectral tails (assumed to be present for frequencies greater than the maximum model frequency) into high model frequencies. The average spectrum frequency then turns out to be relatively high as evidenced in the average frequency plots.

TABLE 2

MODEL PARAMETERS FOR CASES VI A-B

Domain	E-W extent: 270-1030 km; N-S extent: 700-1700 km; parametric hurricane model (Ross, 1976) used for boundary and initial conditions; perfect absorption for outgoing rays
Grid	40 km, regular
Rays	30 parallel rays, ea. dir., 25-35 km ray-point spacing
Time Step	1200 sec
Run Time	24 hr
Eye of Storm	Initial position: (650, 1400) km for VI A, (650, 104) km for VI B; final position: (650, 1400) km
Frequencies (20)	.045, .05, ..., .08, .09, ..., .16, .18, .225, .25 hz
Directions (24)	0, 15, 30, ..., 330, 345°

The General Custer plot, Fig. 67, gives some impression of the overall model response to the vortical wind field.

G. CASE VI B, MOVING HURRICANE

The second model hurricane translated northward at 15 m/s. Otherwise, the wind field and model parameters were the same as for CASE VI A (see Sec. III.G and Table 2). The model domain was really much too small for this case and the boundary conditions imposed via the model by Ross (1976) affect the solution markedly.

The locations of model output points are given in Fig. 64 and all of the plots for the model intercomparison are given in Figs. 85-104. Significant wave heights for this case reach 9-9.5 m, Fig. 85, almost as large as for the previous case, but in the right rear quadrant of the storm. Most of the models give this response, thought to be reasonable by hurricane aficionados, but field data from a fast-moving hurricane are required for confirmation. Mean frequencies, Fig. 86, drops as low as 0.1 hz, roughly, and, as explained in Sec. III.F, the relatively high mean frequencies near the boundaries are artifacts of the way in which the source terms are treated. (Actually these high means may be quite reasonable.) Spectra with wild directional properties contribute to the messy appearance of the General Custer diagram, Fig. 87. The clear vortical pattern seen in CASE VI A, Fig. 85, is no longer apparent.

Cardone found the present case very inconclusive when he compared different model outputs because of the many confounding variables (Hasselmann et al., 1982). Growth rates for the spectrum and directional relaxation are the key contributors to the model response. Simulating hurricanes is a very practical business and future testing of the model(s) should include applications using real wind and wave data from severe storms.

H. CASE VII, DIRECTIONAL RELAXATION

The large differences among model results for the previous cases pointed toward a simpler test of directional relaxation. In an unbounded ocean area subject to a constant wind of 20 m/s and with a partially- or fully-developed wave spectrum the wind shifts instantly by 90°; the ensuing model response is termed directional relaxation in a so-called 'duration-limited' (rather than 'fetch-limited') situation. T. Barnett of SIO conducted part of this test using the same spectral balance as in the full wave prediction model but without the propagation algorithm and all its trappings.

Figures 105-108 describe the response of the simplified model following a 90° wind shift from N to W when the sea is fully-developed ($f_m = f_u = .13$ g/U). Below the spectral peak the spectrum changes very slowly (Figs. 107-8). It retains high energy and with nowhere to propagate dominates the growth after t=0. Indeed, mean frequency f_0 remains small and variance E (Fig. 105) drops so the nonlinear terms (proportional to $E^3 f_0^8$) cannot grow the mid range frequency very rapidly. In real life the 'swell' would propagate away and much more rapid growth would be evident at mid-range frequencies. Between f_p and f_{cut} the spectrum is immediately limited by the local fully developed sea. Thus the energy between N and E is lost and that between N and W is limited by a $\cos^2$ distribution centered on W. The growth in this frequency range is slow, however, for the reasons noted above. Above f_{cut} the local sea is set to that expected for the local wind, i.e., a subgrid growth mechanism is used. The implied turning times, are crudely in balance with theory: "Swell" is unaltered; waves traveling slower than the wind respond rapidly. We could add some finesse in dealing with the mid range frequencies but choose not to until we see the results of some more realistic tests

(on actual data). As indicated with regard to CASE V (Sec. III. E), the drop in variance associated with the effects of the limiting spectrum on mid frequencies is probably unrealistic, and some rethinking is necessary. On the bright side, real winds take several hours to switch by 90° at least over the spatial scales resolved by most wave prediction models, so this problem may be less severe than it first appears.

III. SUMMARY

We have applied the discrete nonlinear spectral model to seven ideal cases of wind wave generation as part of an international study to compare wave prediction models. Our primary purpose is to save the results, which were costly in personnel and computer time, for future reference. The primary areas we see for model improvement and rethinking are subgrid scale growth mechanisms and appropriate behavior under rapidly changing winds. A study using actual wind and wave data seems essential in order to really assess model capabilities. The effects of uncertainties in the wind field should be quantified, also.

IV. REFERENCES

Allender, J.H., T.P. Barnett, and M. Lybanon (1982). An Improved Spectral Model for Ocean Wave Prediction. Proc. Symp. on Wave Dyn. and Radio Probing of the Ocean Surface, Eds. K. Hasselmann and O. Phillips. Plenum Press.

Allender, J.H. and M. Lybanon (1982). A Discrete Nonlinear Spectral Model for Ocean Wave Prediction; Description of Computer Program. NORDA Tech. Note NTN 147, NSTL Station, MS.

Hasselmann, K. and X Others (1982). The Wave Model Intercomparison Study. Proc. Symp. on Wave Dyn. and Radio Probing of the Ocean Surface, Eds. K. Hasselmann and O. Phillips. Plenum Press.

____________, D. Ross, P. Muller, and W. Sell (1976). A Parametric Wave Prediction Model. J. Phys. Oceanog., 6, 2, 200-227.

Ross, D. (1976). A Simplified Model for Forecasting Hurricane Generated Waves (Abstract). Bull. A.M.S., January, 113.

Snyder, R.L., F.W. Dobson, J.A. Elliot, R.B. Long (1981). Array Measurements of Atmospheric Pressure Fluctuations Above Surface Gravity Waves. J. Fluid Mech., 102, 1-59.

V. TABLE OF MATHEMATICAL NOTATION

Common notation was established for the intercomparison. Symbols used in the text and the figures are summarized below.

(f,θ)	Wave component frequency and direction
(x,y)	Spatial coordinates
g	Acceleration of gravity
t	Time
U	Wind speed (m/s)
θ_U	Wind direction
$F(f,\theta)$	2-D wave spectrum
$F(f)$	1-D wave spectrum, $\int F d\theta$
F_{max}	Maximum value of 1-D or 2-D spectrum, according to context
E	Variance of sea surface displacement, $\iint F df d\theta$
f_U (or f_{PM})	Wind frequency, .13 g/U, peak of Pierson-Moskowitz (PM) spectrum
f_p	Peak frequency of 1-D spectrum (determined by quadratic fit in model)
f_o	Mean frequency of spectrum, $\iint fF\, df d\theta$
$\bar{\theta}$	Mean direction of spectrum, $\tan^{-1}\left[\dfrac{\iint \sin\theta\ F\ df\ d\theta}{\iint \cos\theta\ F\ df\ d\theta}\right]$

$F_{PM}(f_{PM})$	$\alpha g^2 (2\pi)^{-4}(f_{PM})^{-5} e^{-5/4}$, with $\alpha = .0081$, i.e., peak of PM spectrum
E_{PM}	$\alpha g^2 (2\pi f_{PM})^{-4}(1/5)$, i.e., variance of PM spectrum
$\tilde{f}$	f/f_p
f^*	$U_\star f/g$, $U_\star = 0.855$ m/s with $U_{10} = 20$ m/s
t^*	gt/U
x^*	$fx^2/U_\star$
E^*	$g^2 E/U_\star^4$

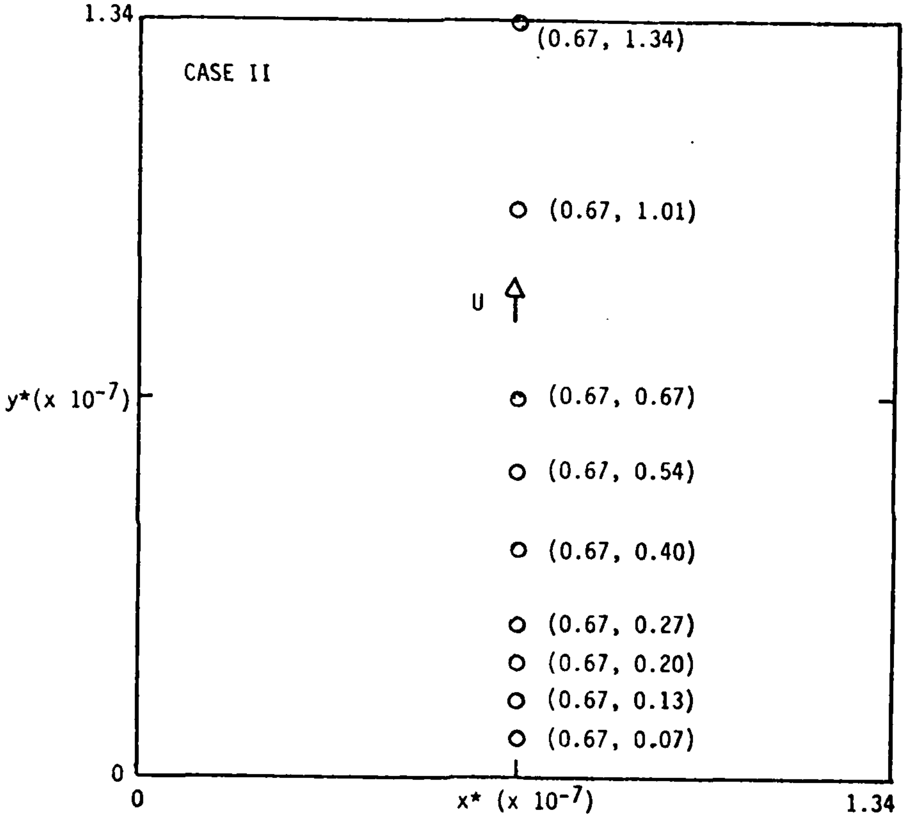


Figure 1. Model output points, Case II

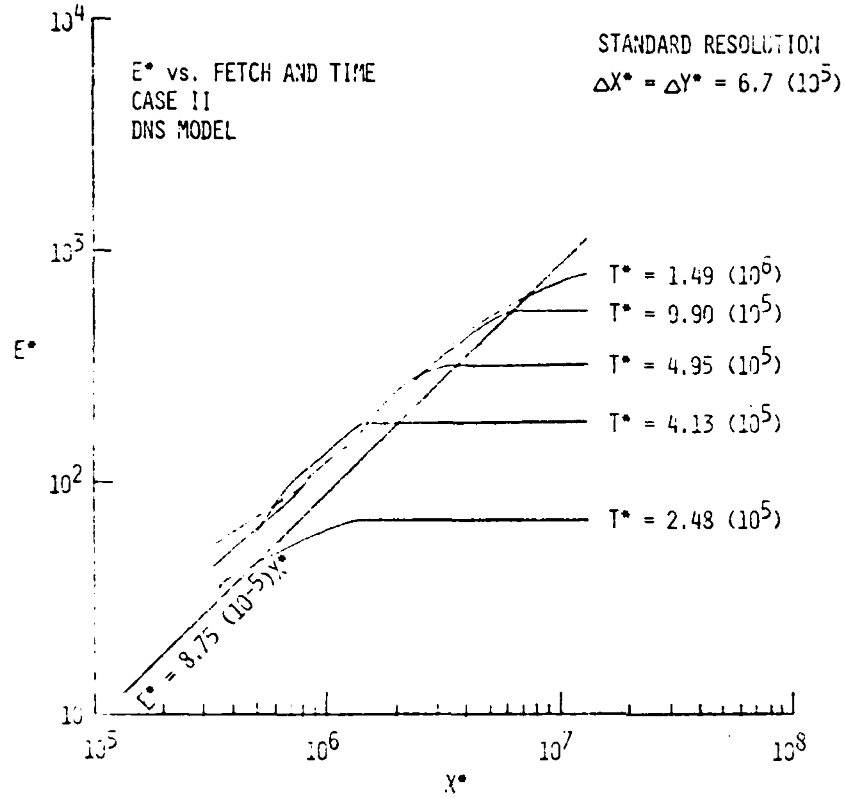


Figure 2.
Model results, Case II

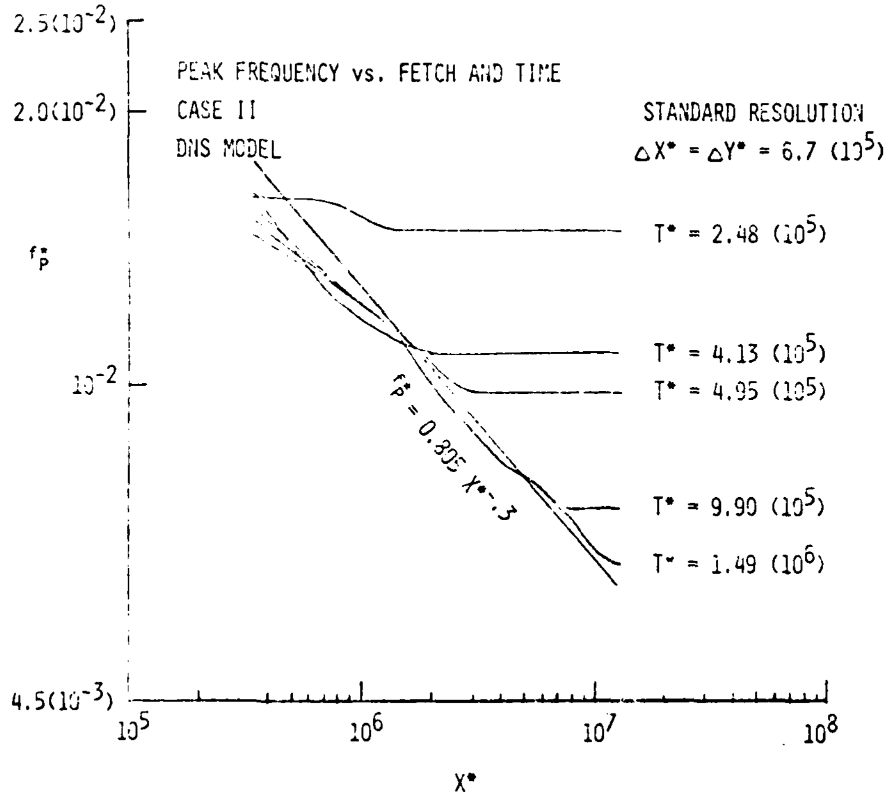


Figure 3.
Model results, Case II

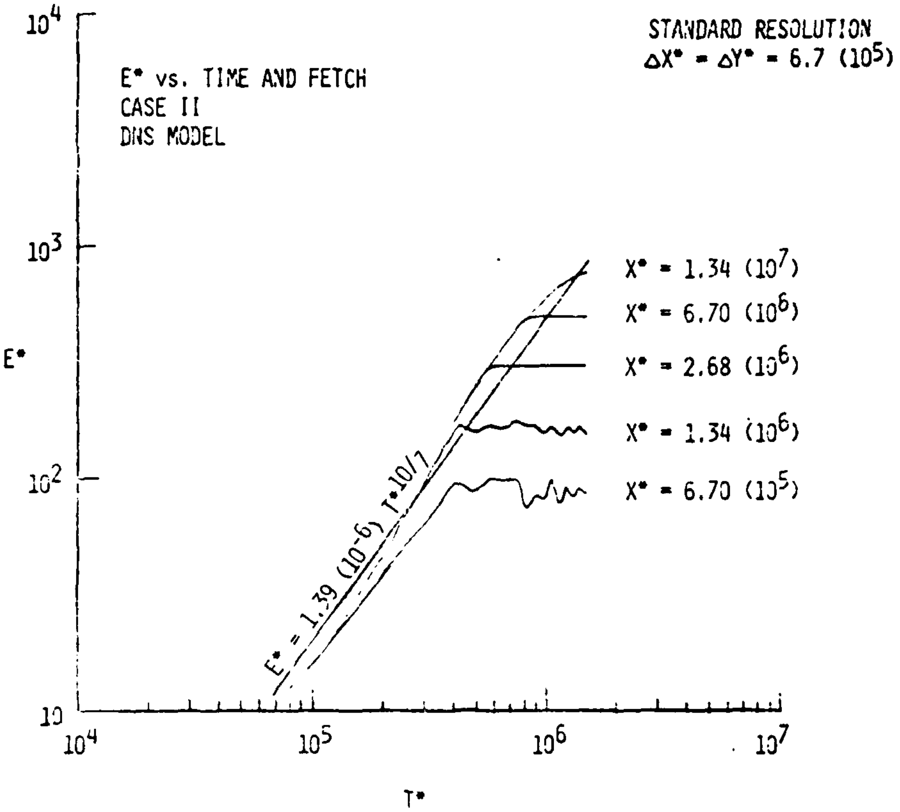


Figure 4.
Model results, Case II

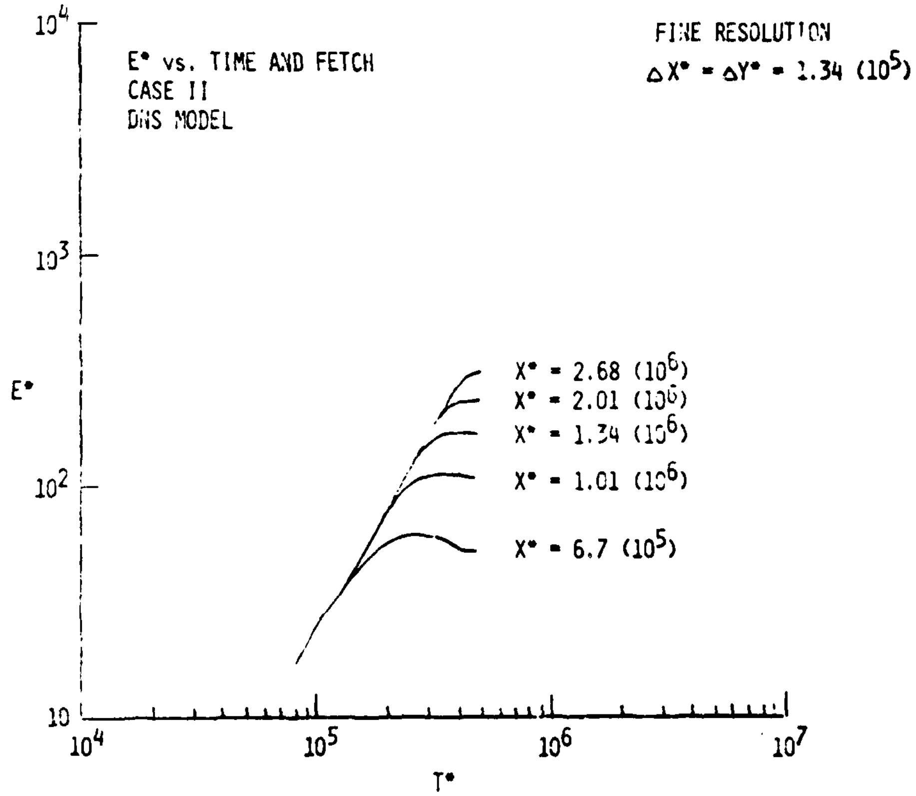


Figure 5.
Model results, Case II

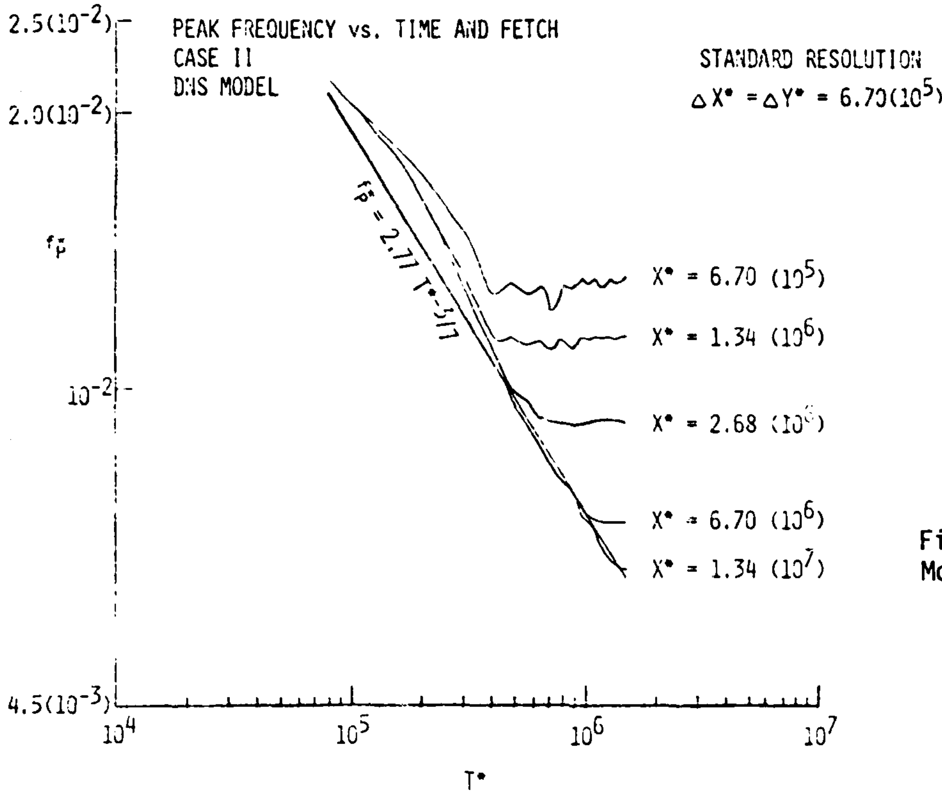


Figure 6.
Model results, Case II

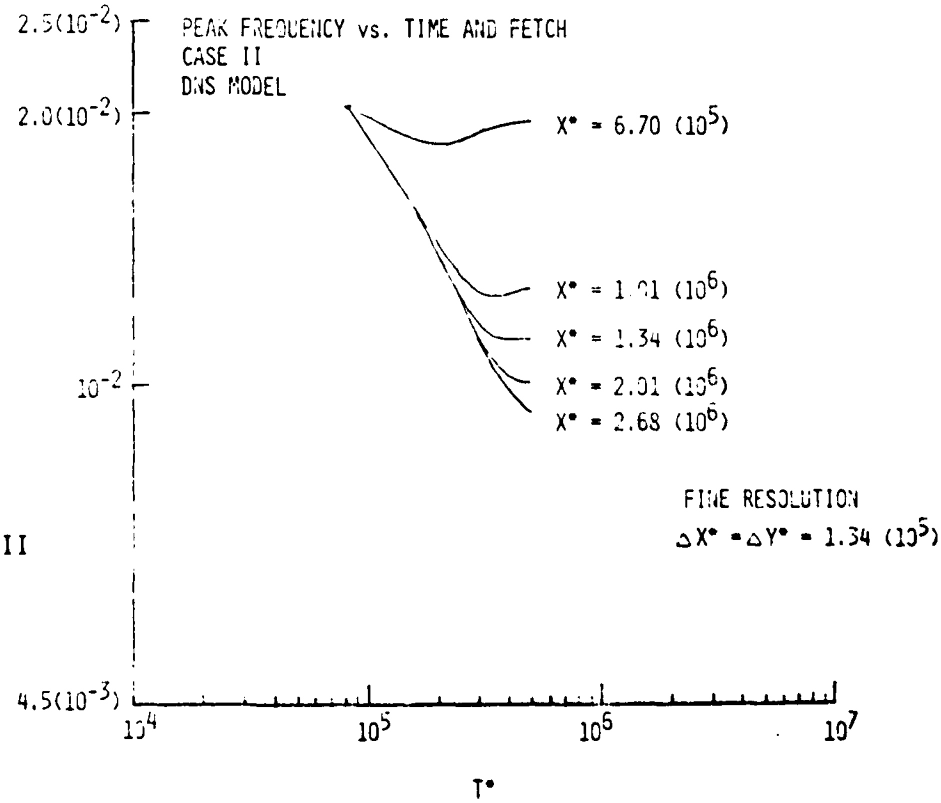


Figure 7.
Model results, Case II

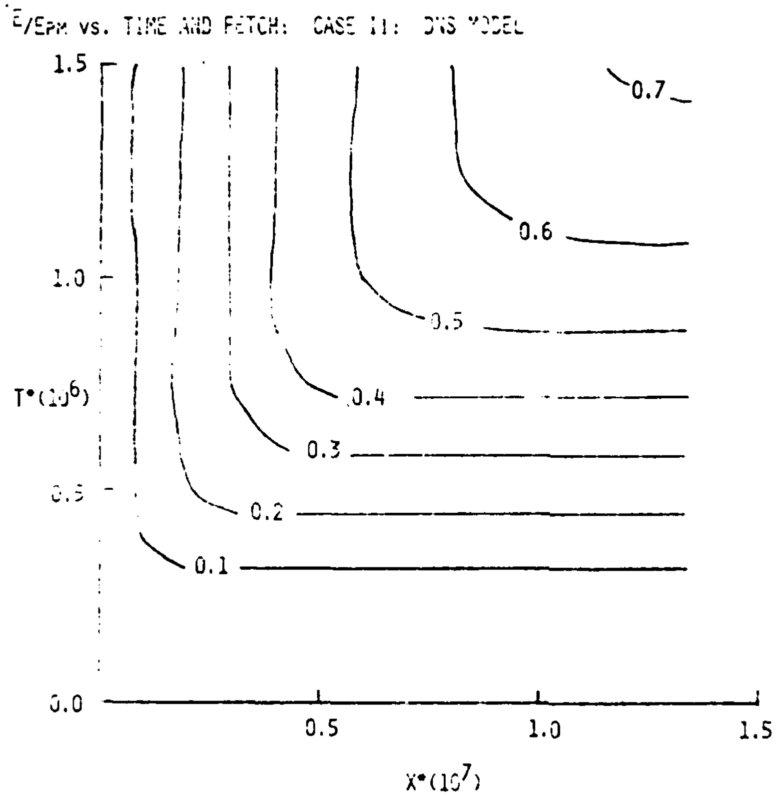


Figure 8.
Model results, Case II

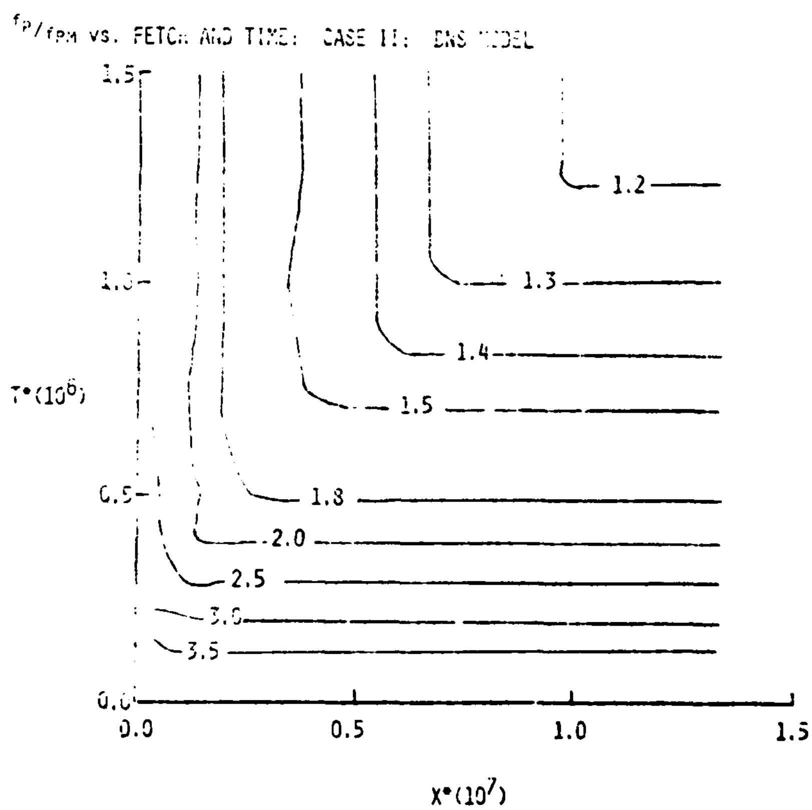


Figure 9.
Model results, Case II

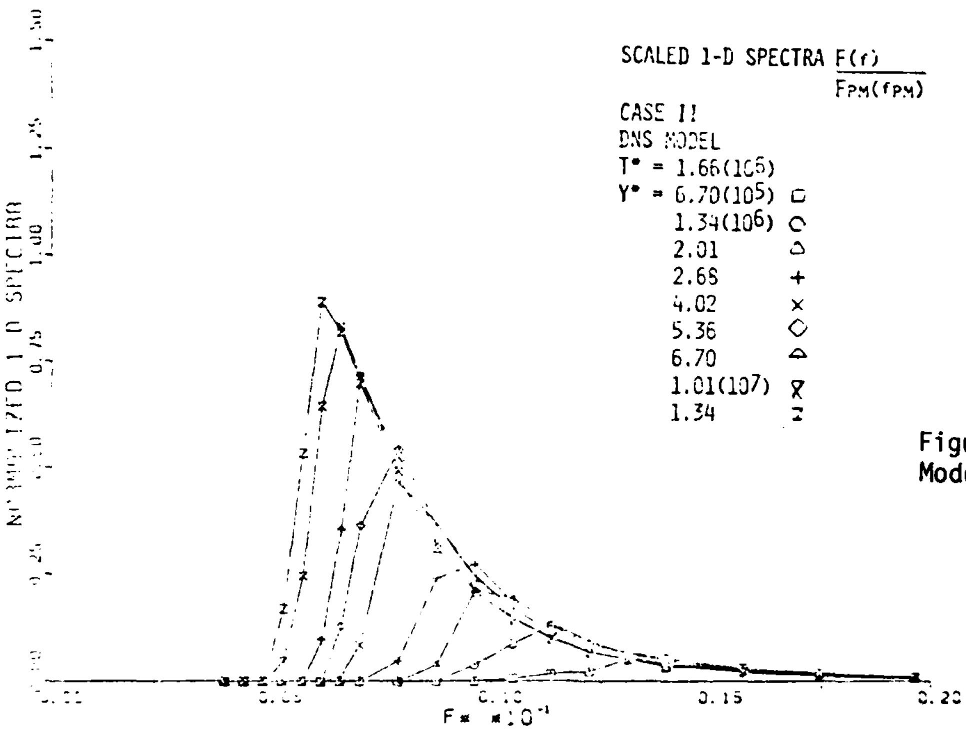


Figure 10.
Model results, Case II

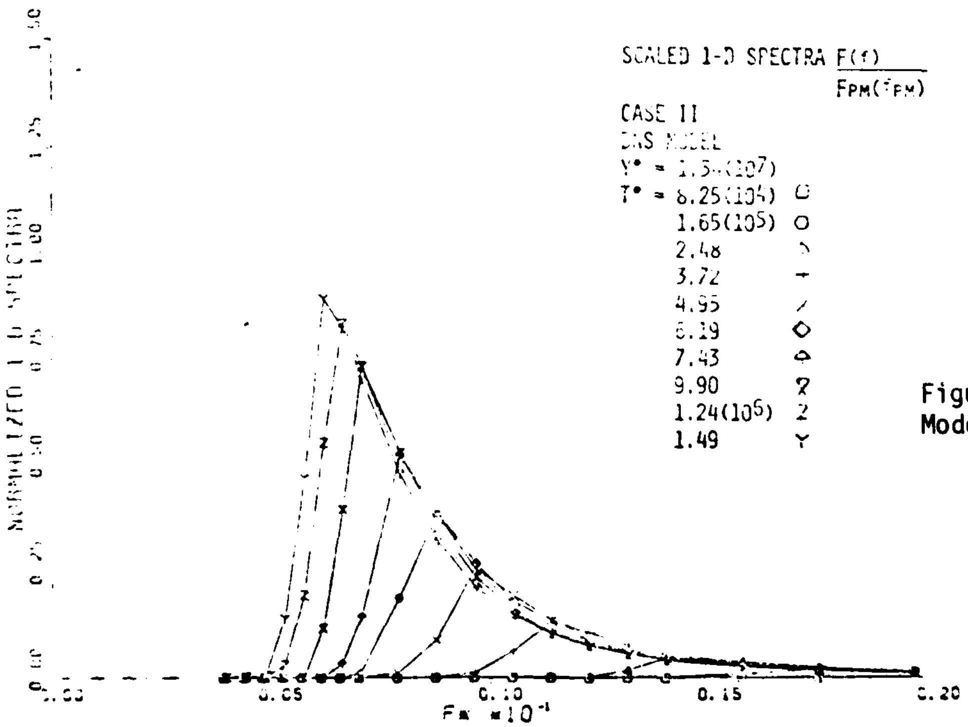


Figure 11.
Model results, Case II

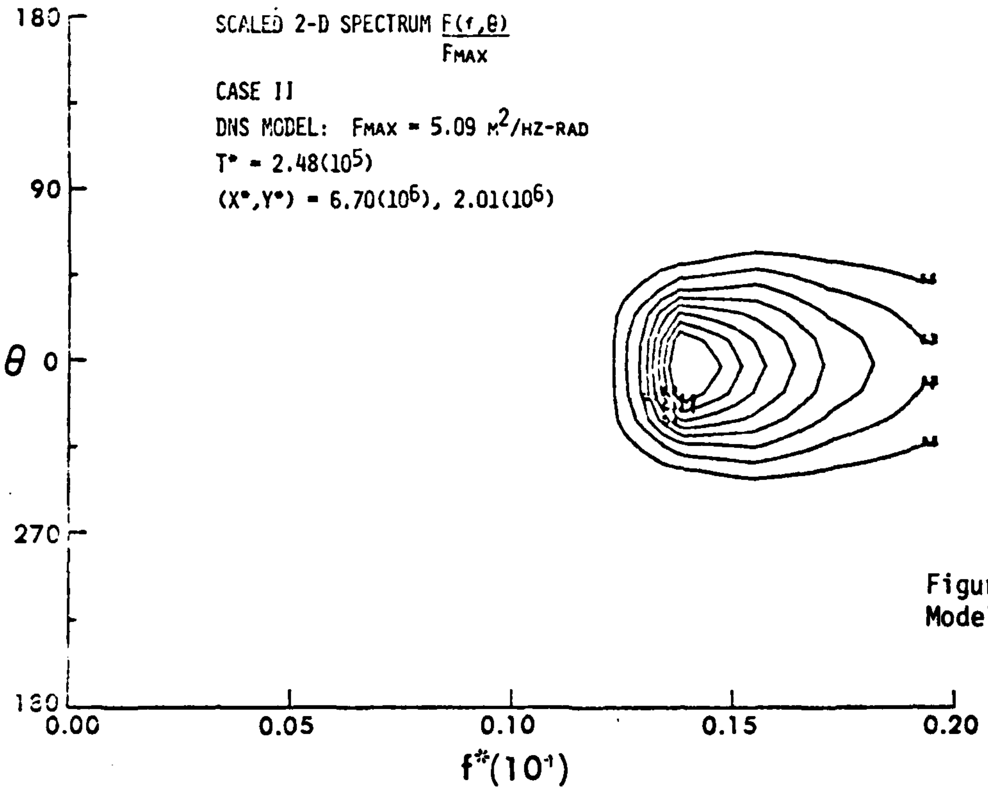


Figure 12.
Model results, Case II

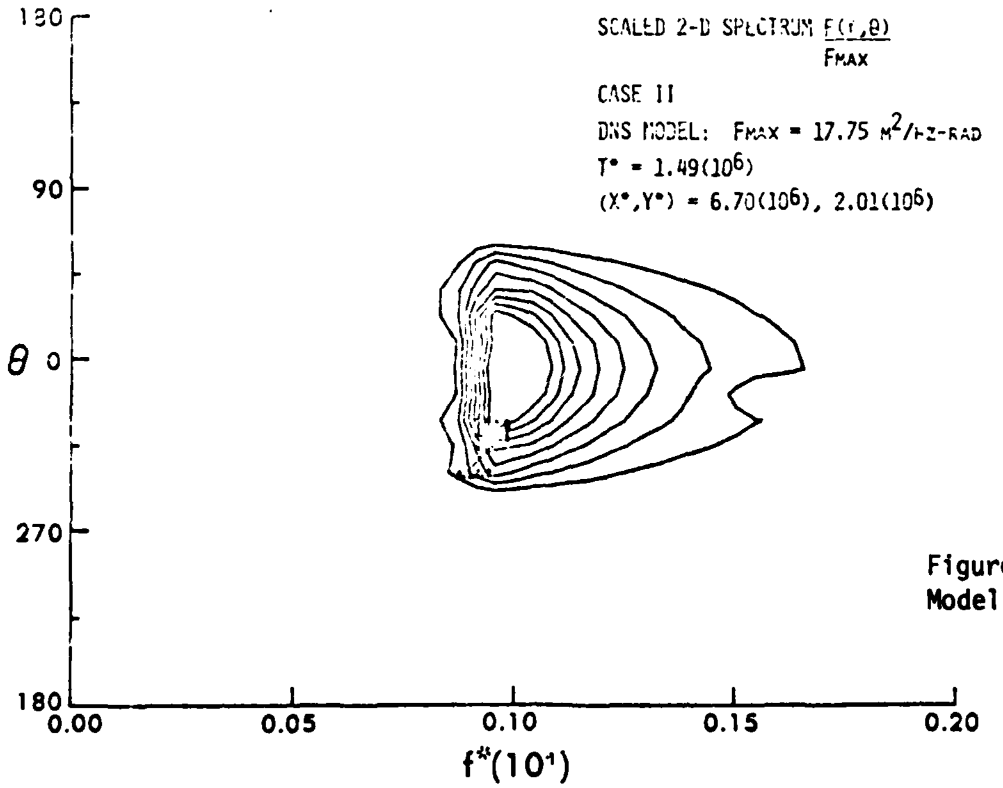


Figure 13.
Model results, Case II

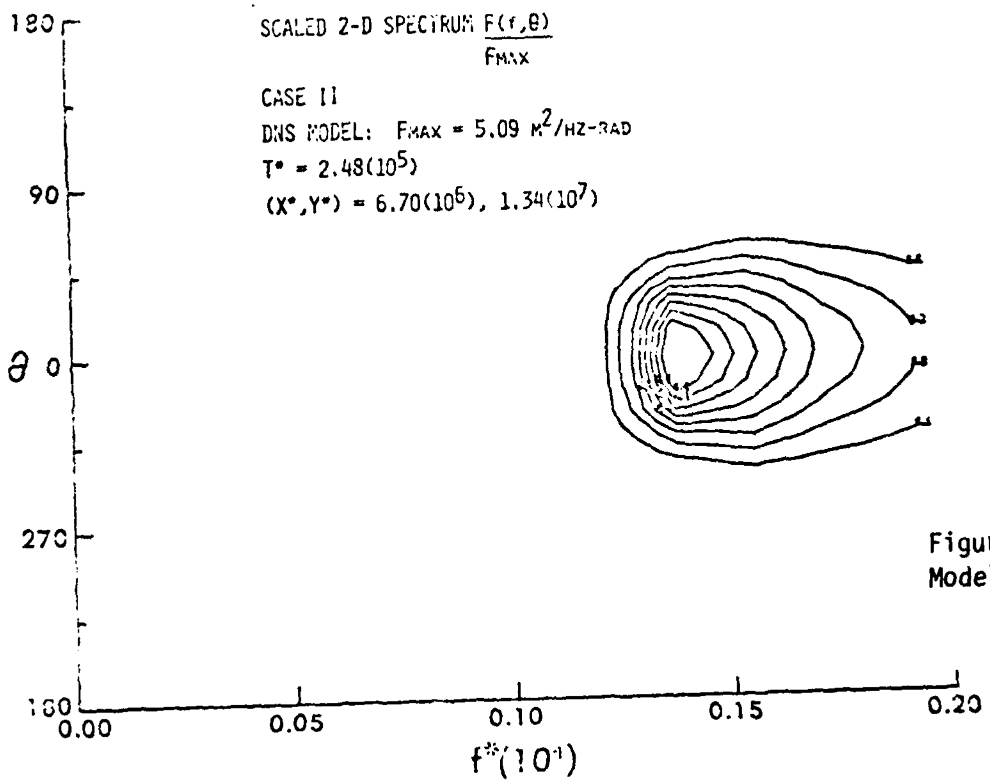


Figure 14.
Model results, Case II

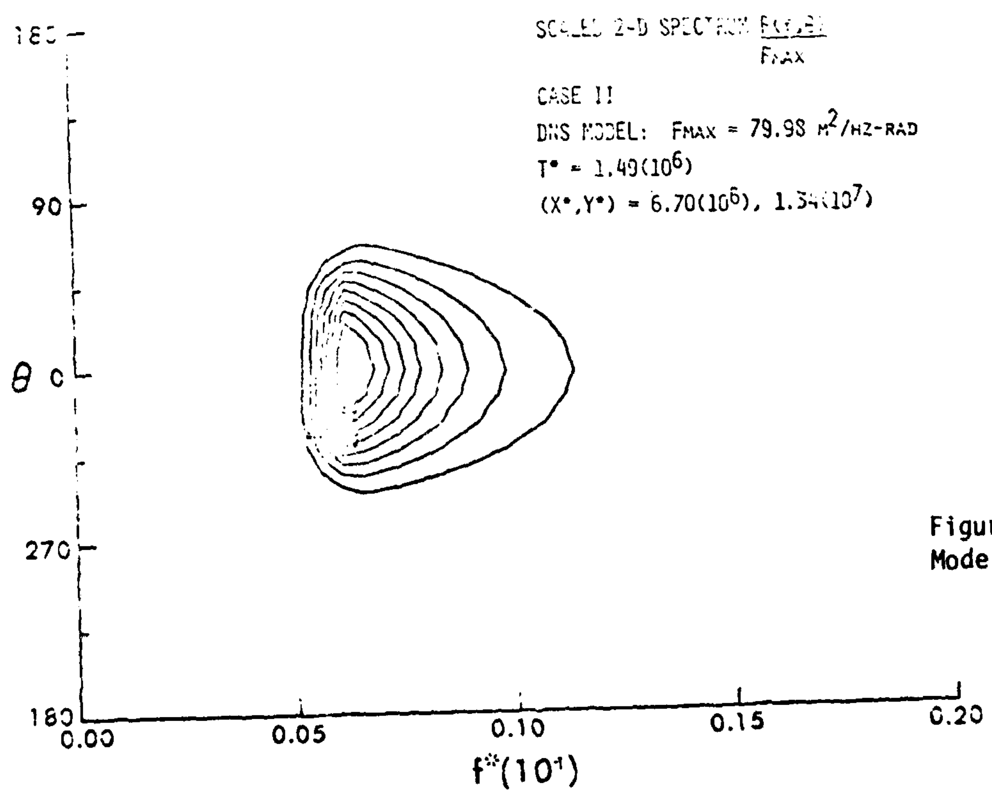


Figure 15.
Model results, Case II

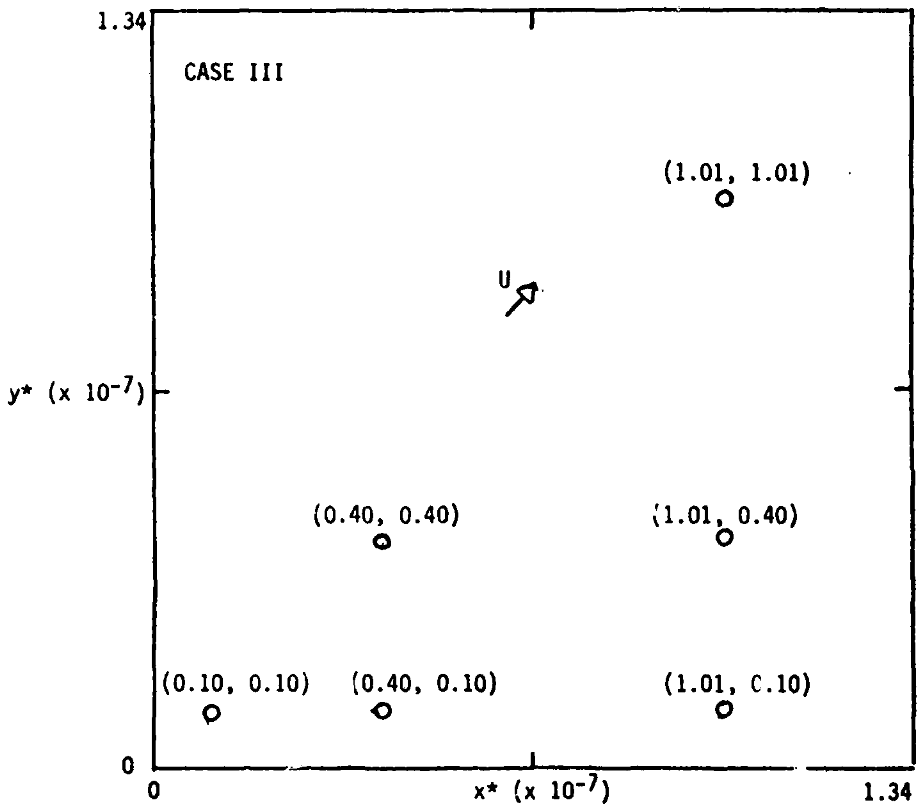


Figure 16. Model output points, Case III

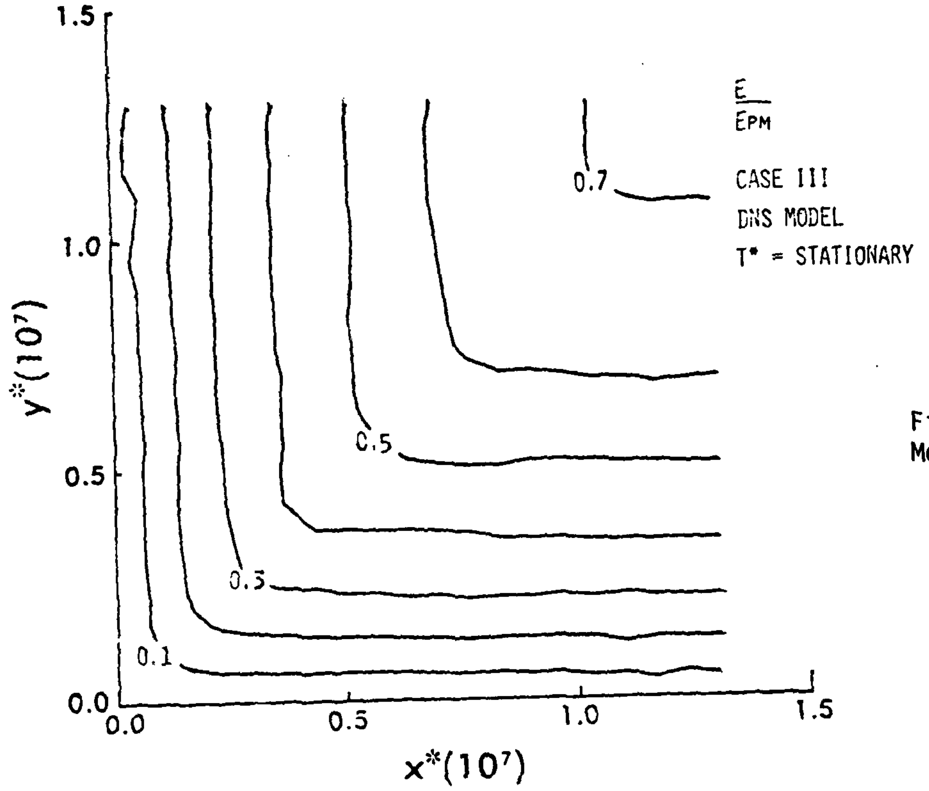


Figure 17.
Model results, Case III

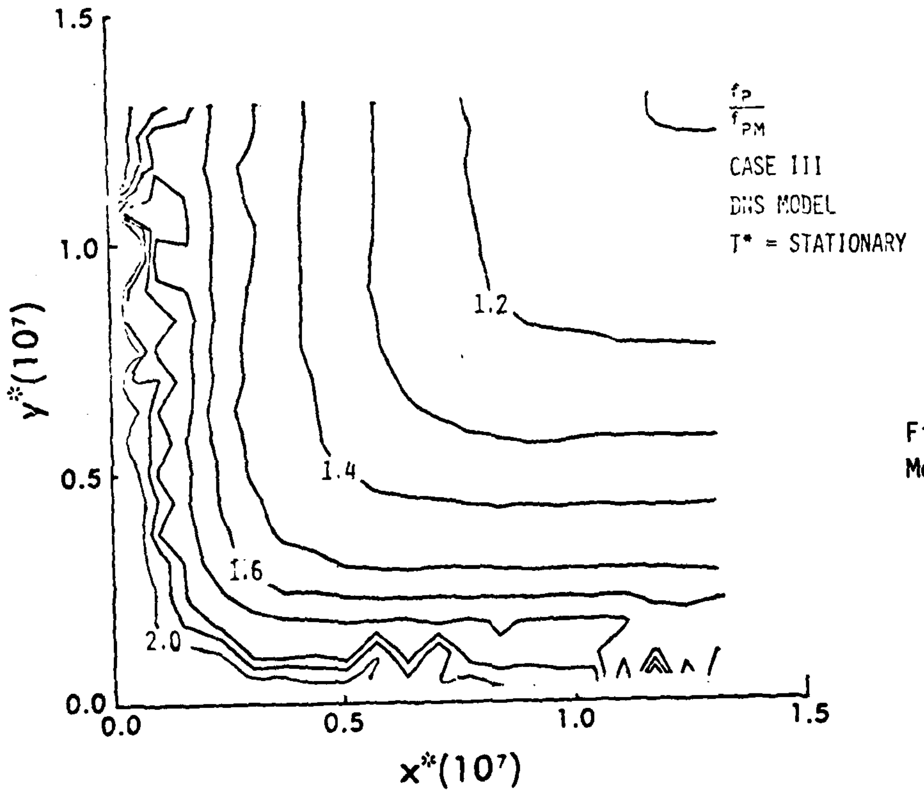


Figure 18.
Model results, Case III

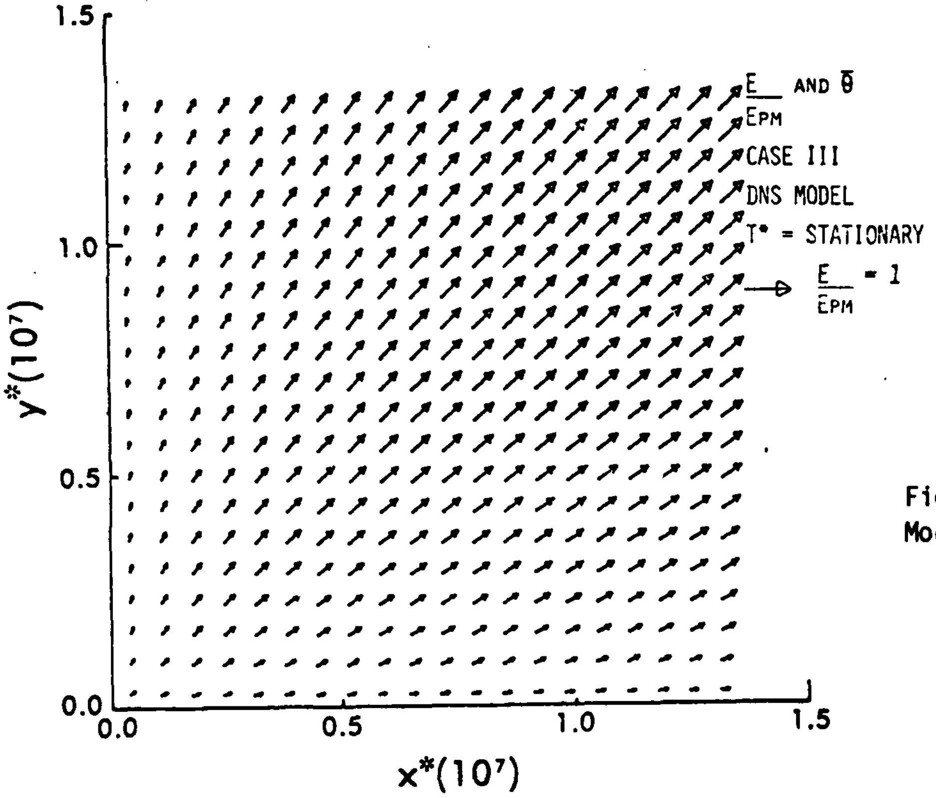


Figure 19.
Model results, Case III

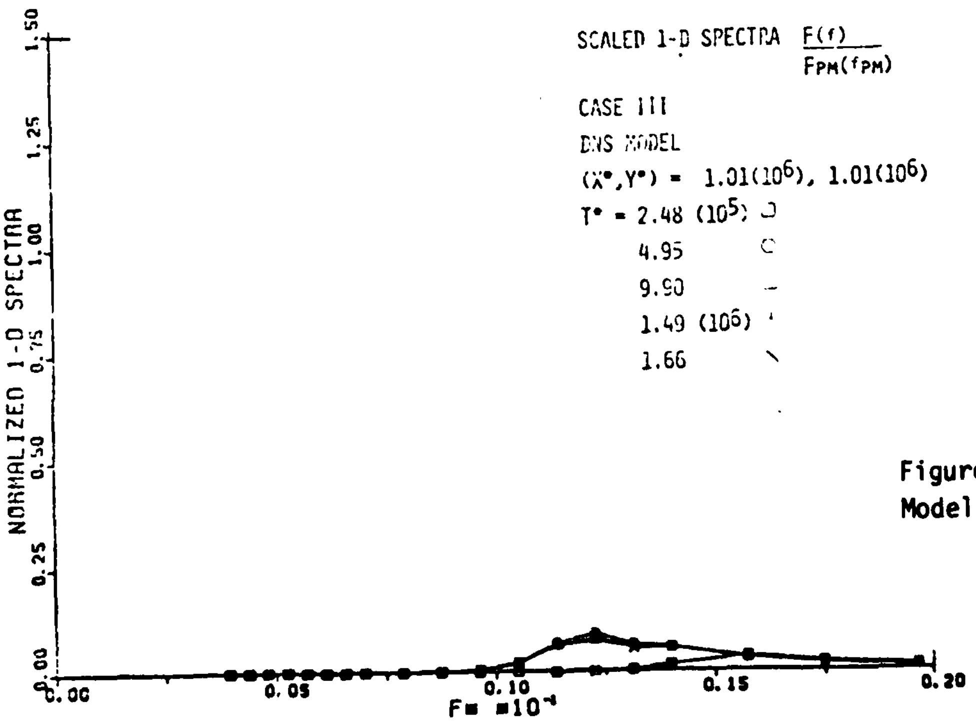


Figure 20.
Model results, Case III

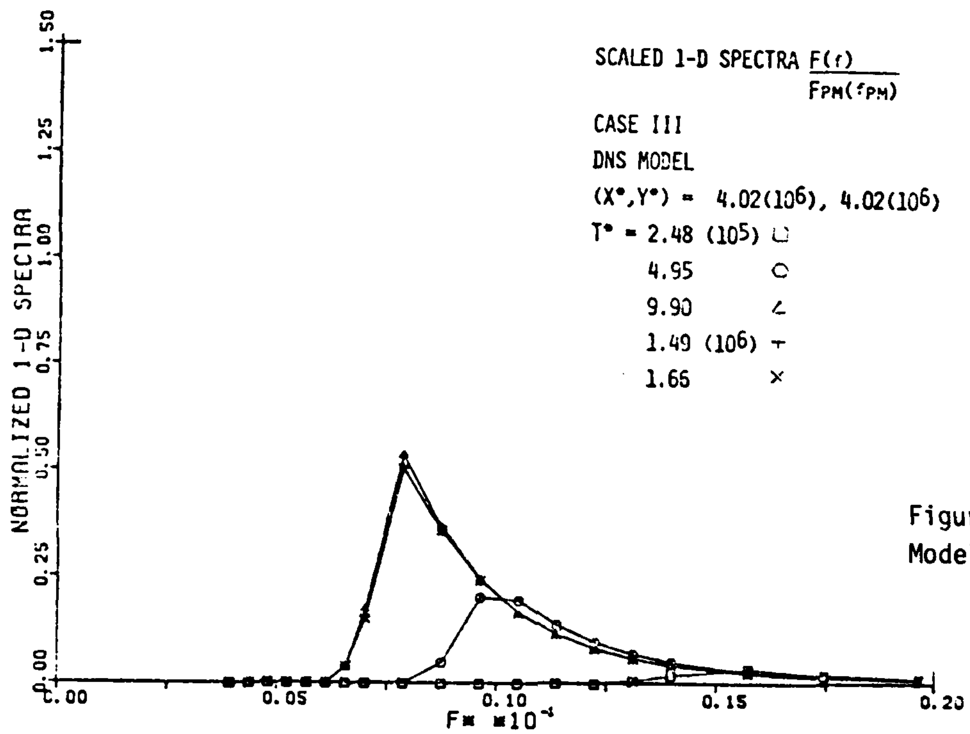


Figure 21.
Model results, Case III

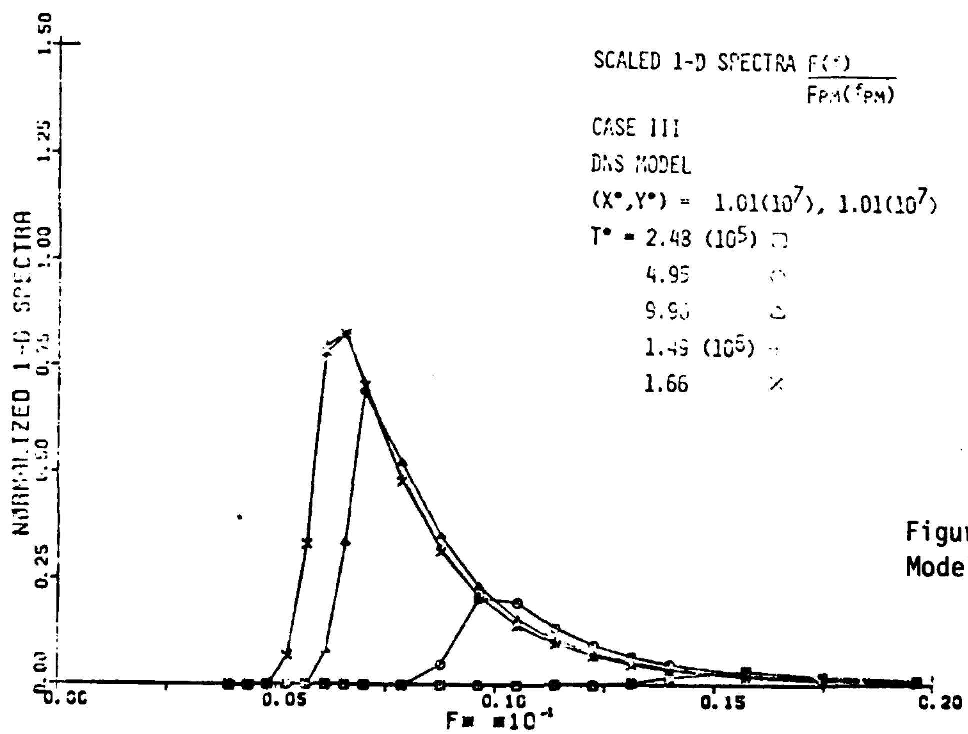


Figure 22.
Model results, Case III

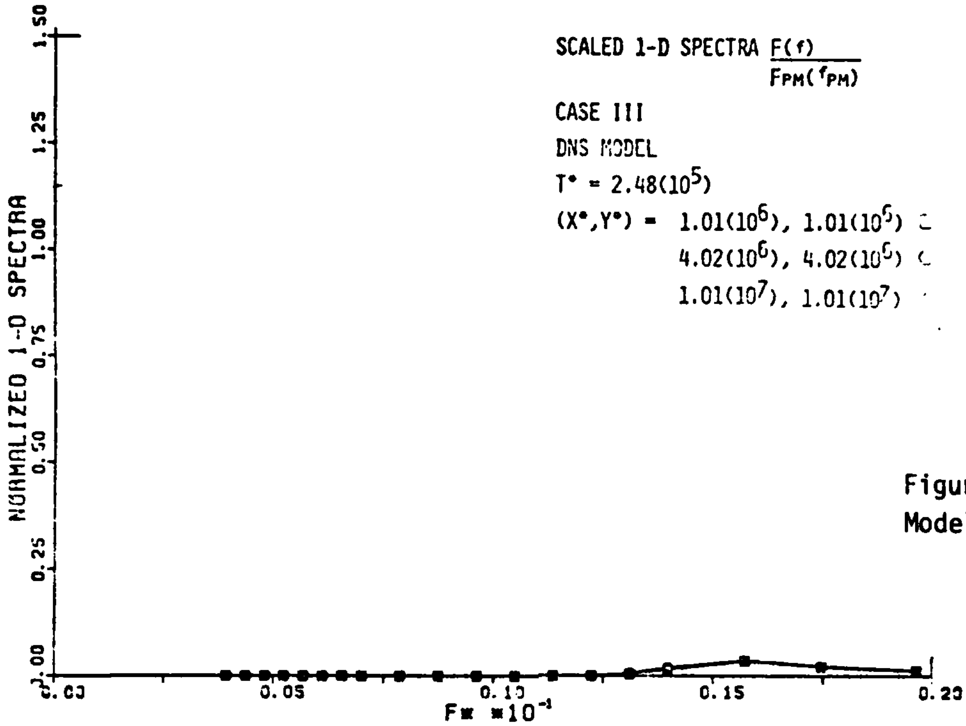


Figure 23.
Model results, Case III

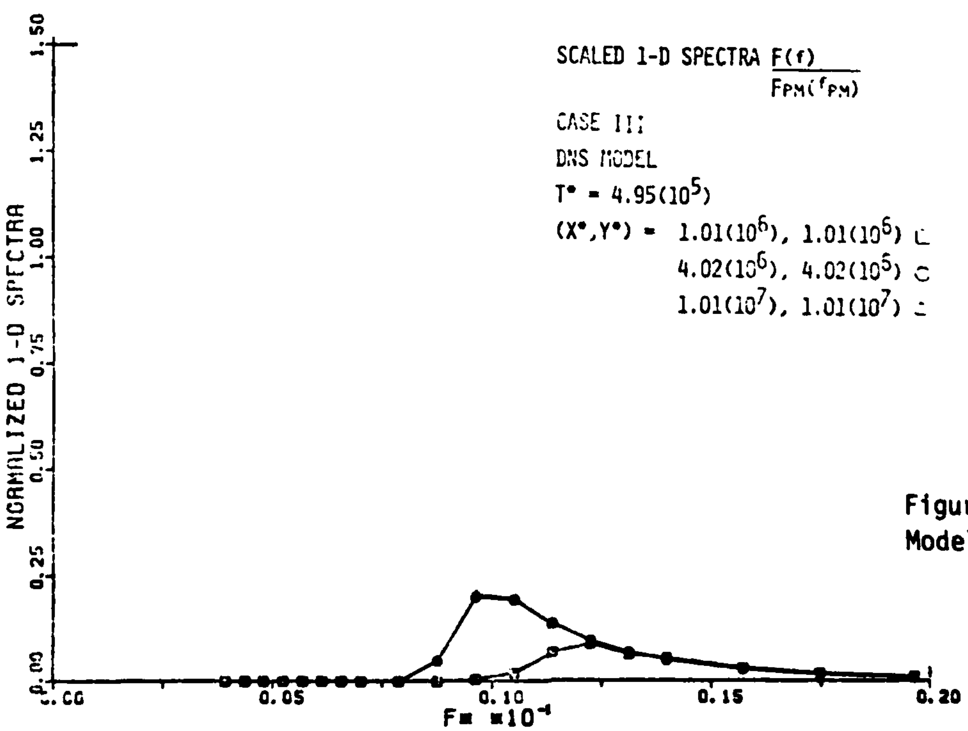


Figure 24.
Model results, Case III

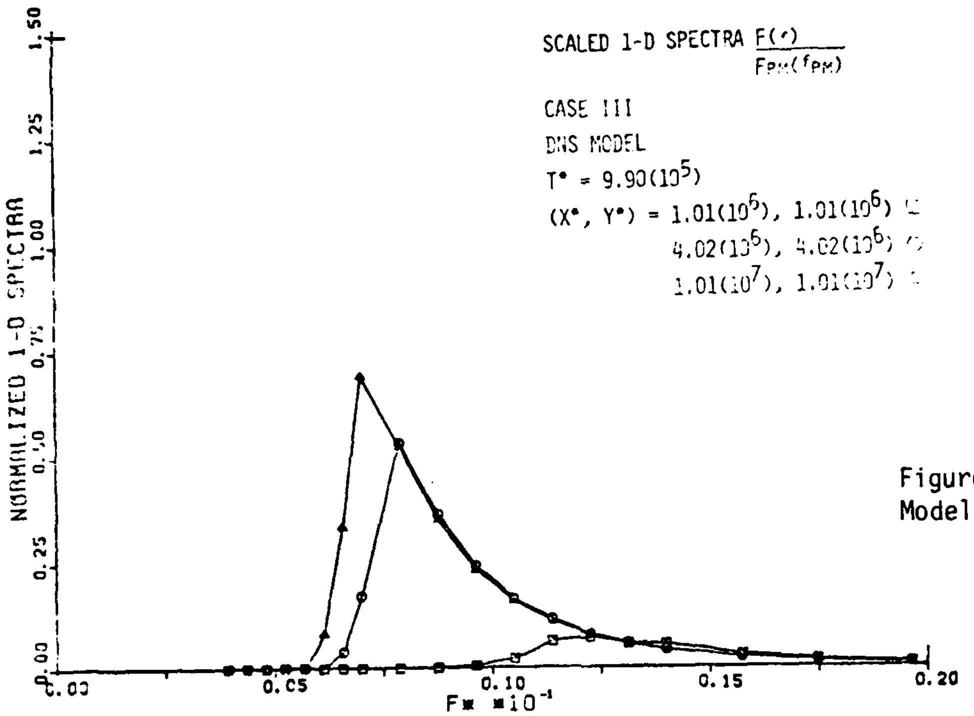


Figure 25.
Model results, Case III

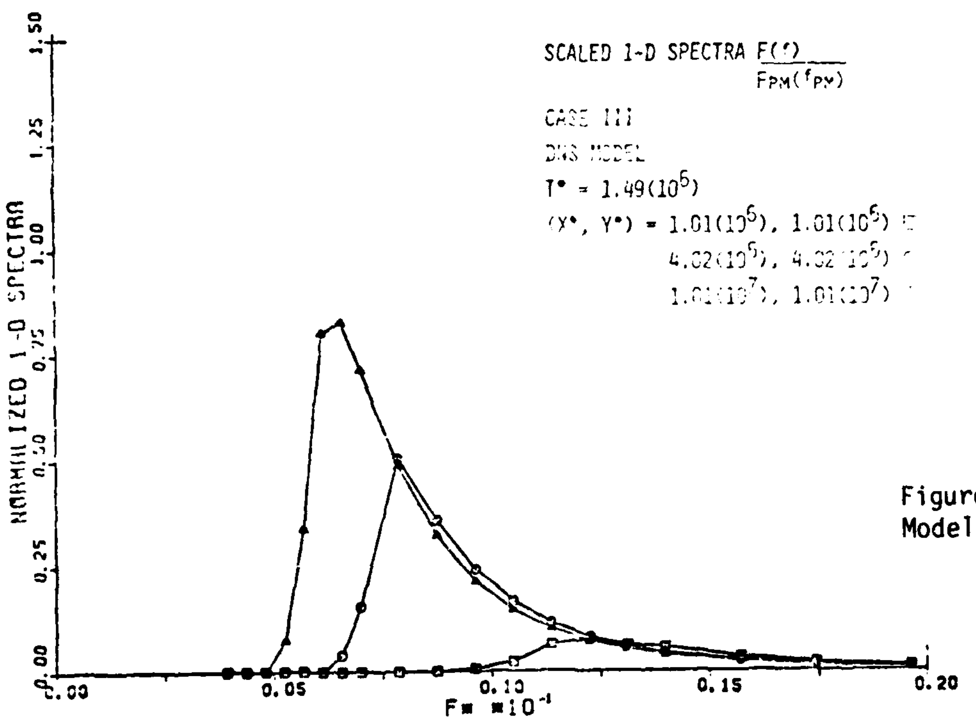


Figure 26.
Model results, Case III

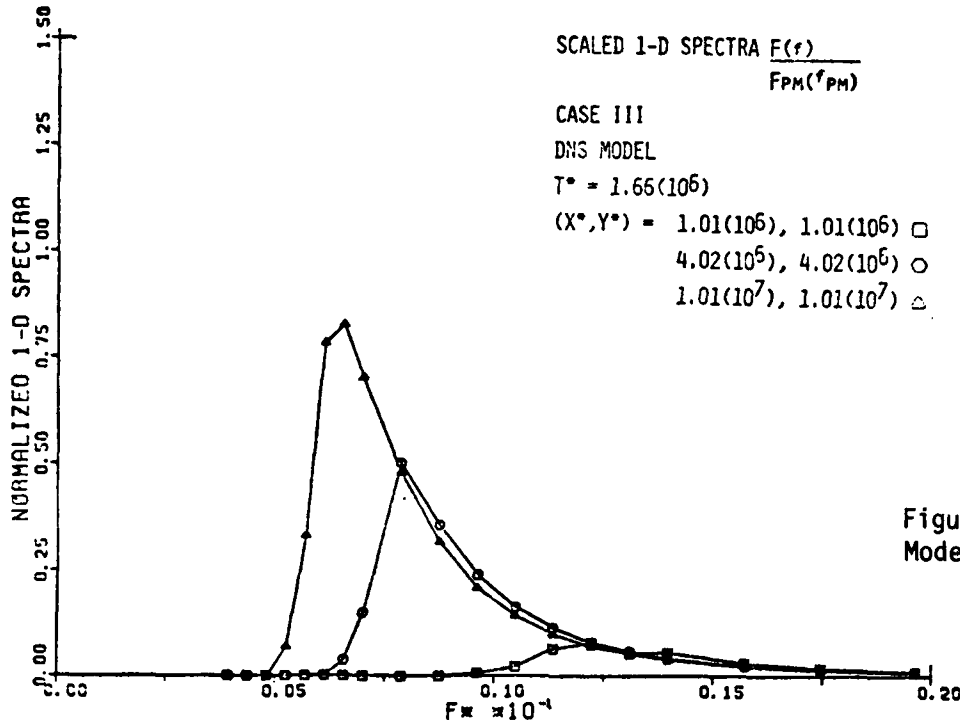


Figure 27.
Model results, Case III

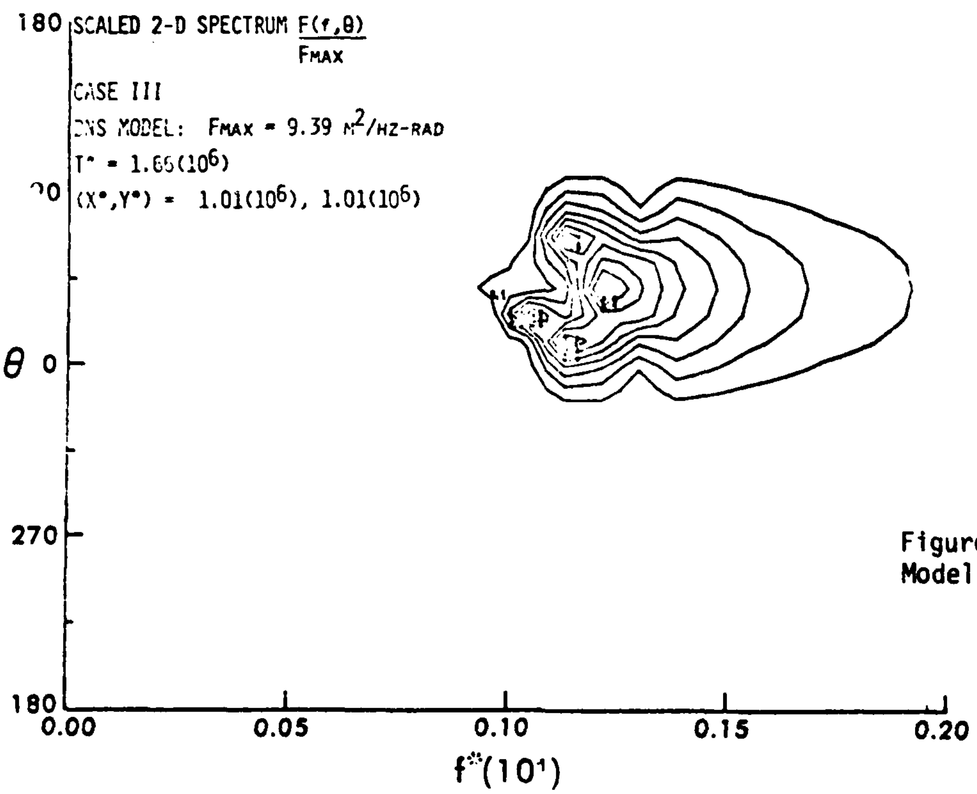


Figure 28.
Model results, Case III

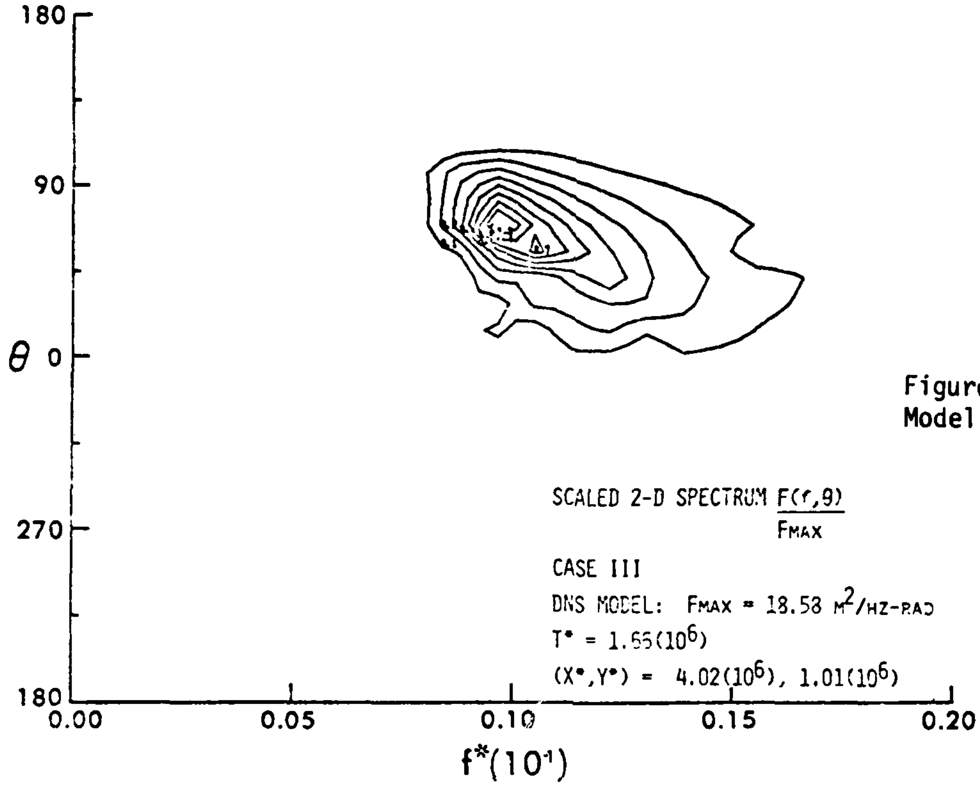


Figure 29.
Model results, Case III

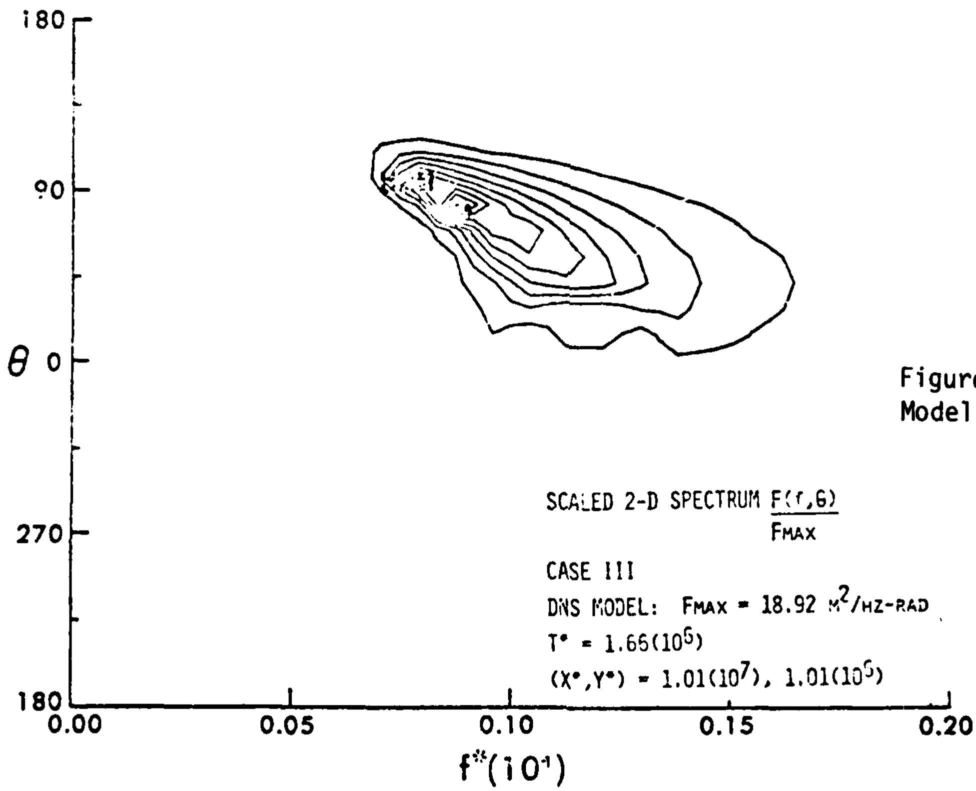


Figure 30.
Model results, Case III

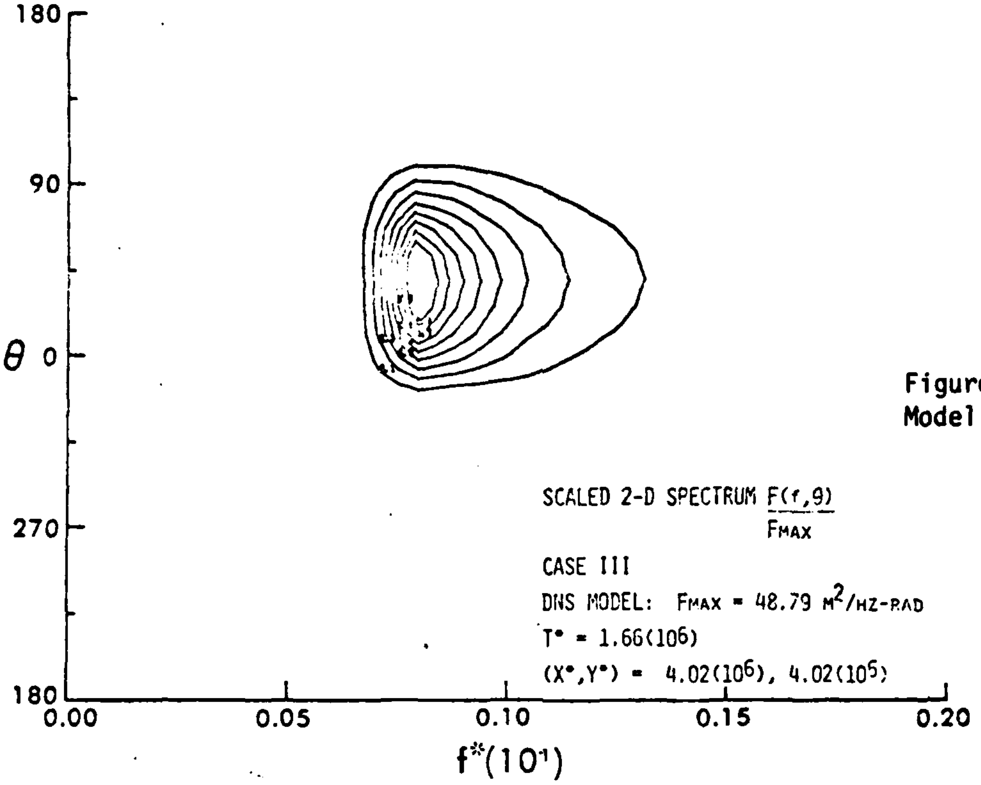


Figure 31.
Model results, Case III

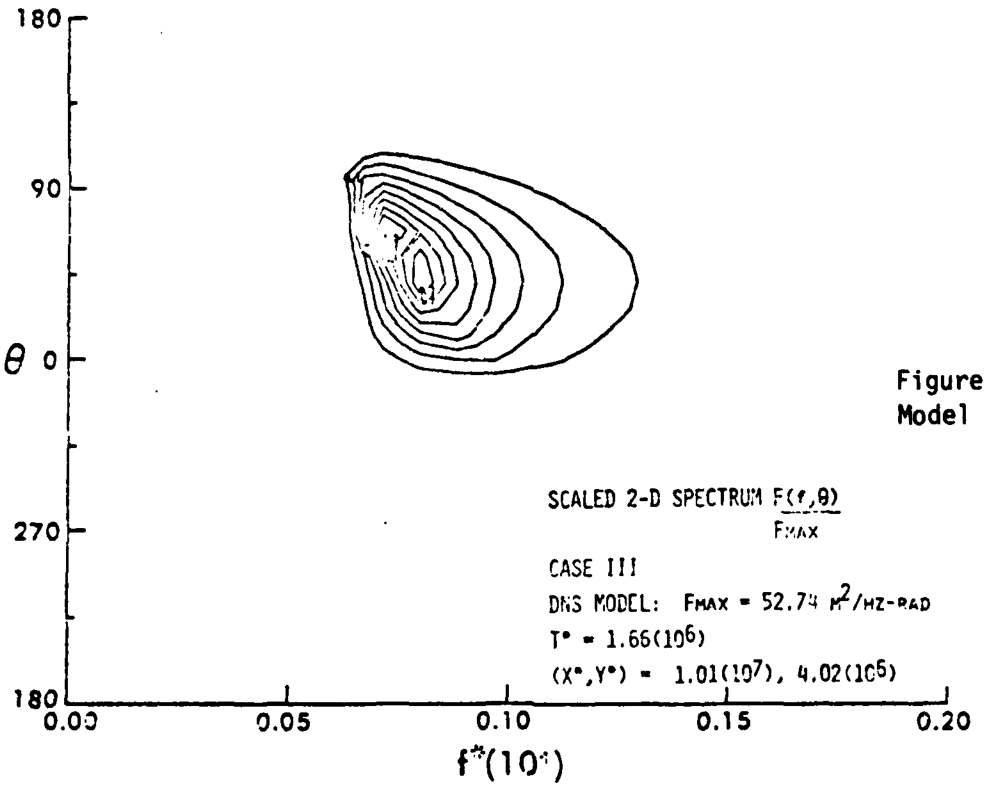


Figure 32.
Model results, Case III

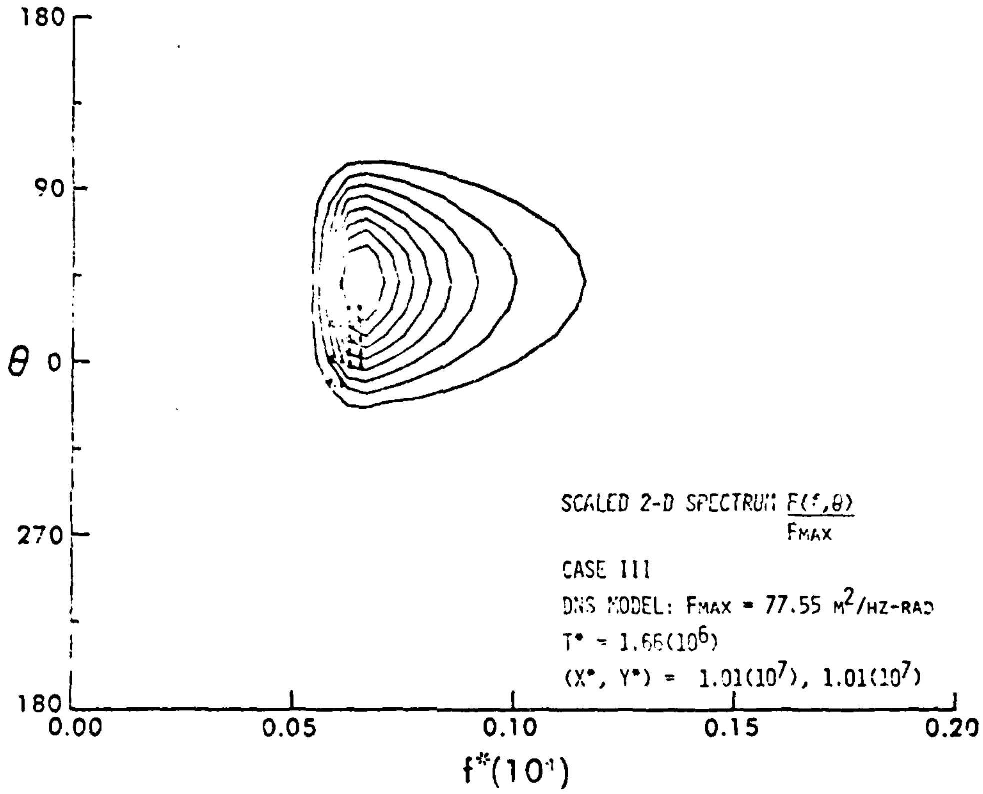


Figure 33. Model results, Case III

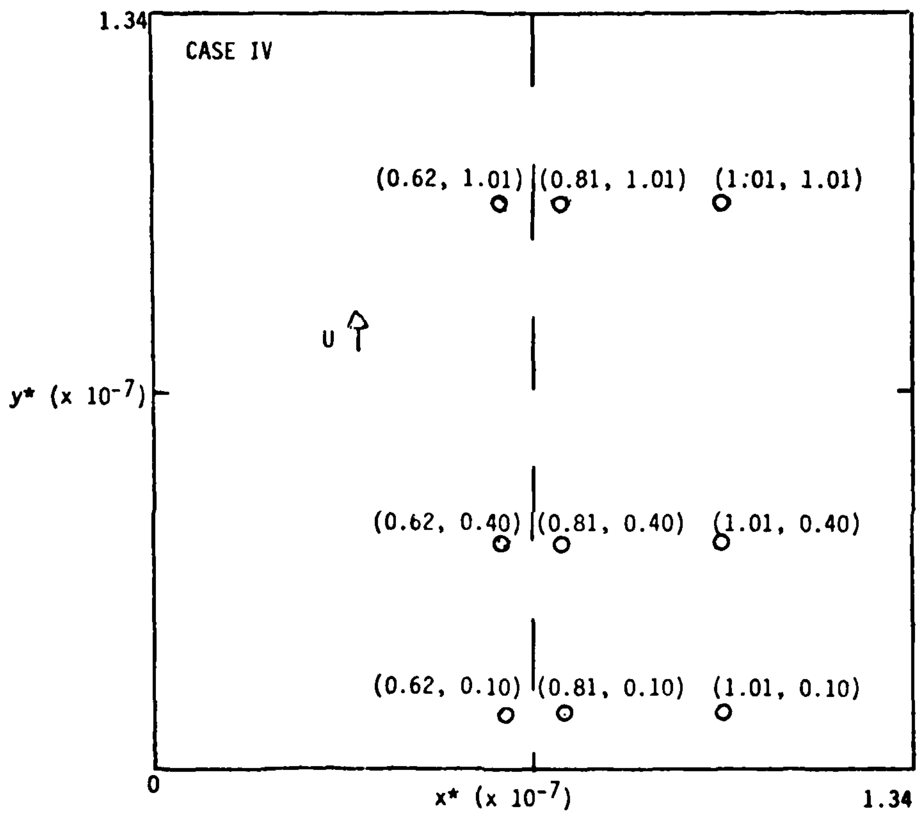


Figure 34. Model output points, Case IV

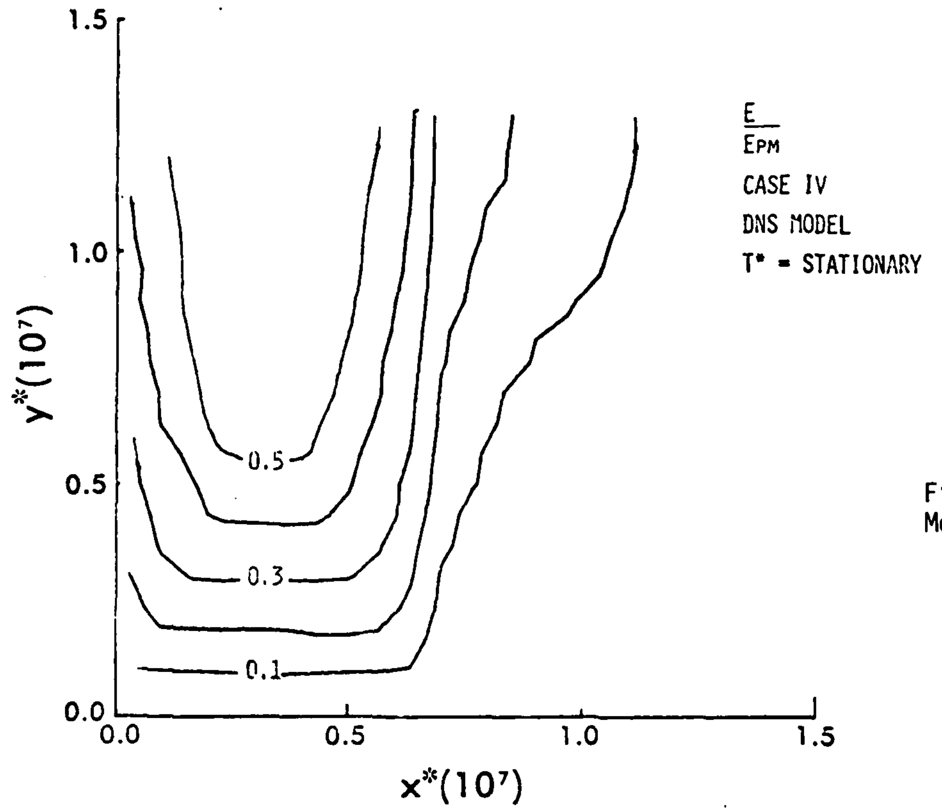


Figure 35.
Model results, Case IV

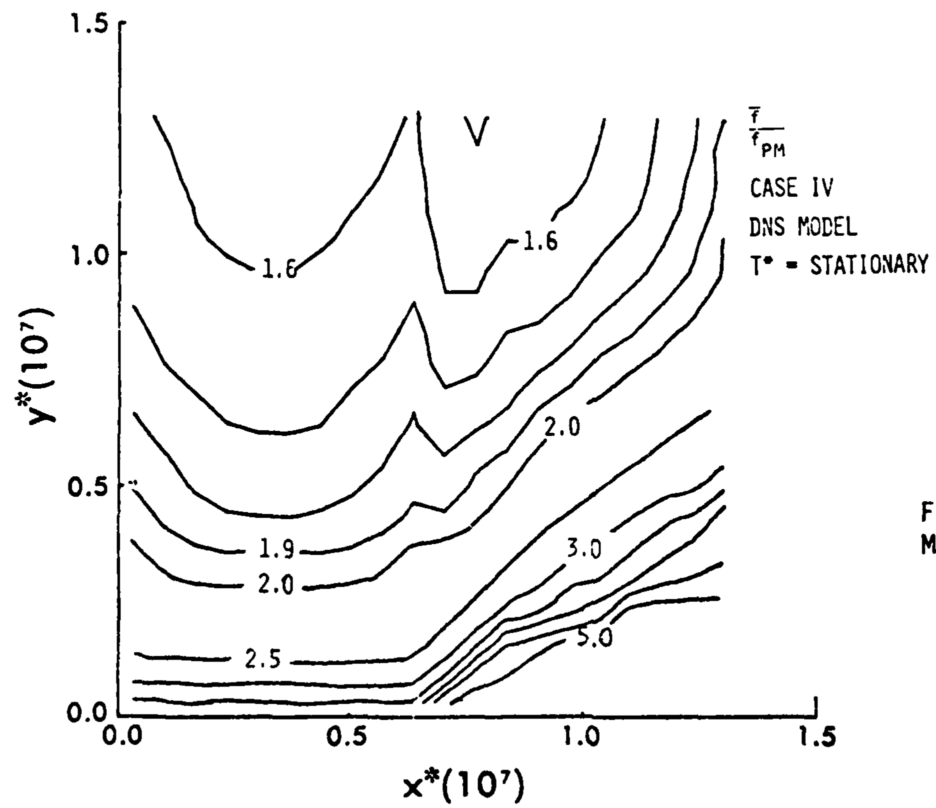


Figure 36.
Model results, Case IV

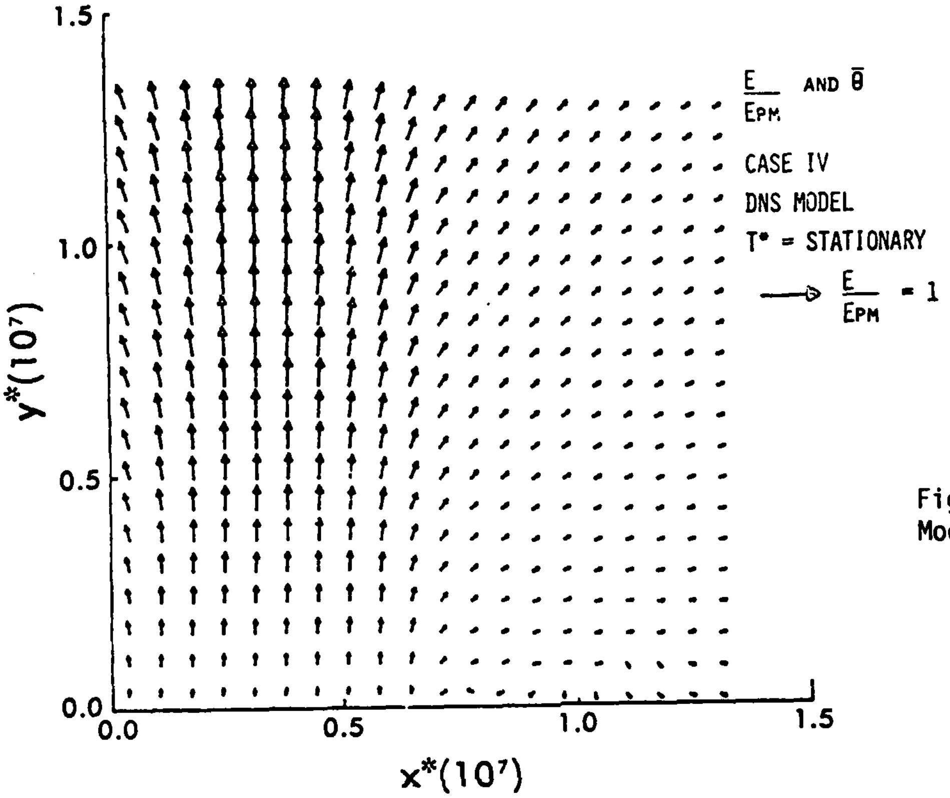


Figure 37.
Model results, Case IV

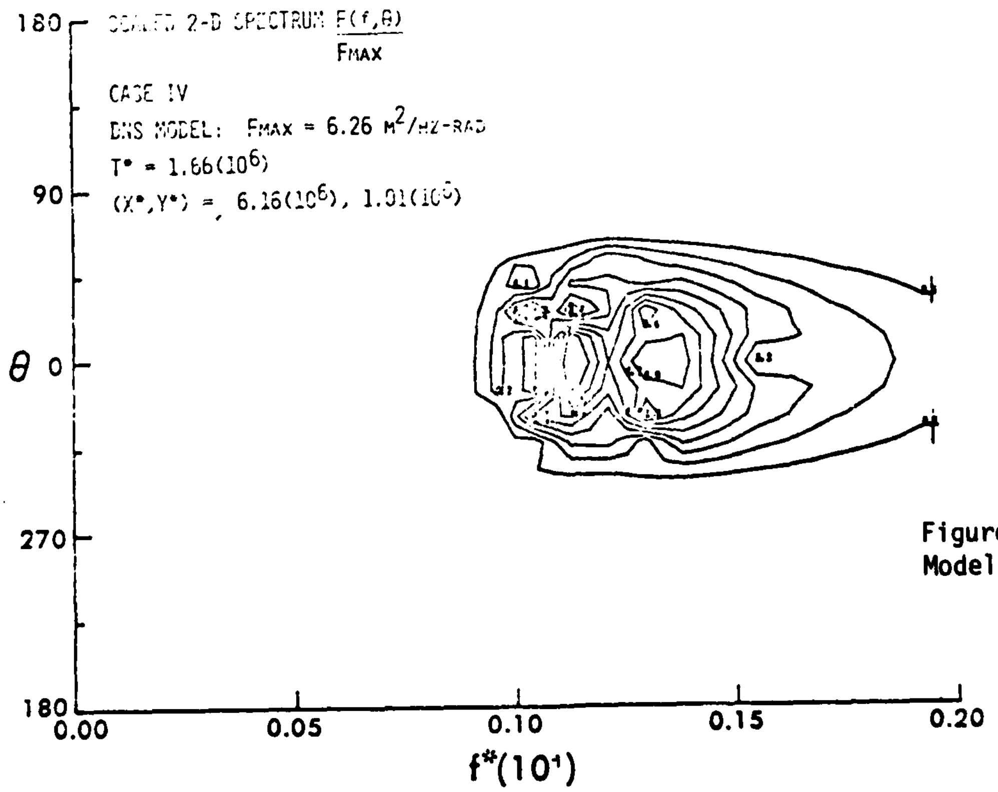


Figure 38.
Model results, Case IV

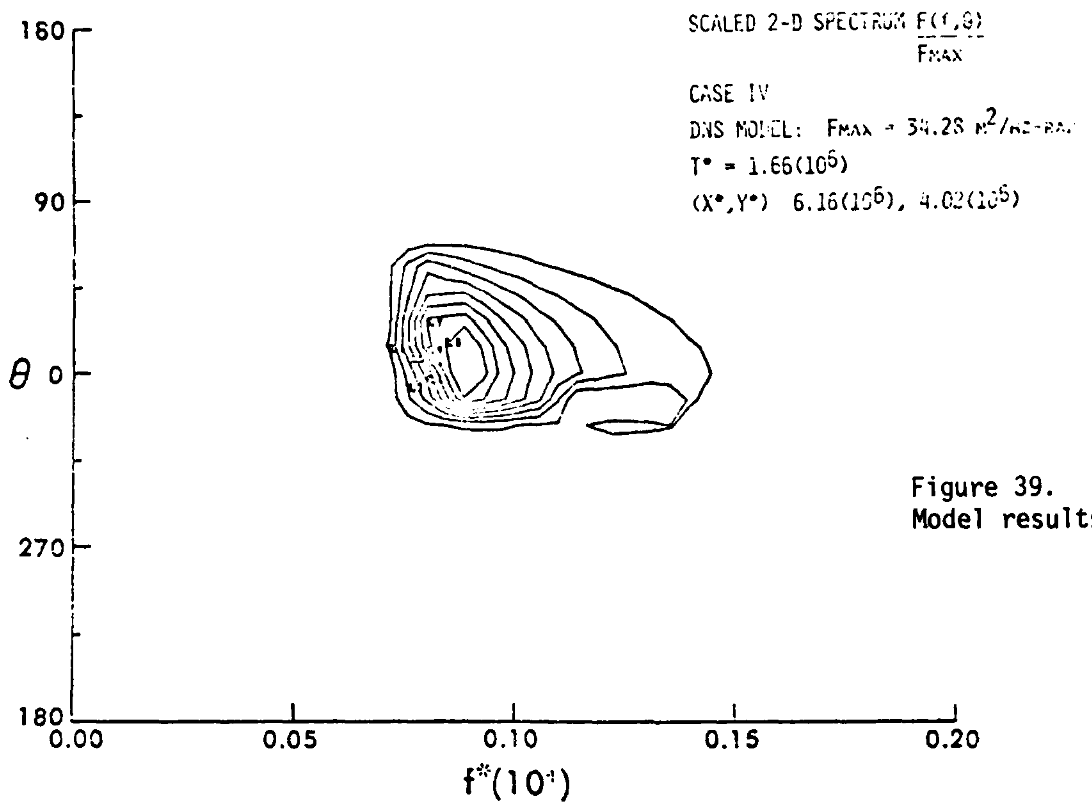


Figure 39.
Model results, Case IV

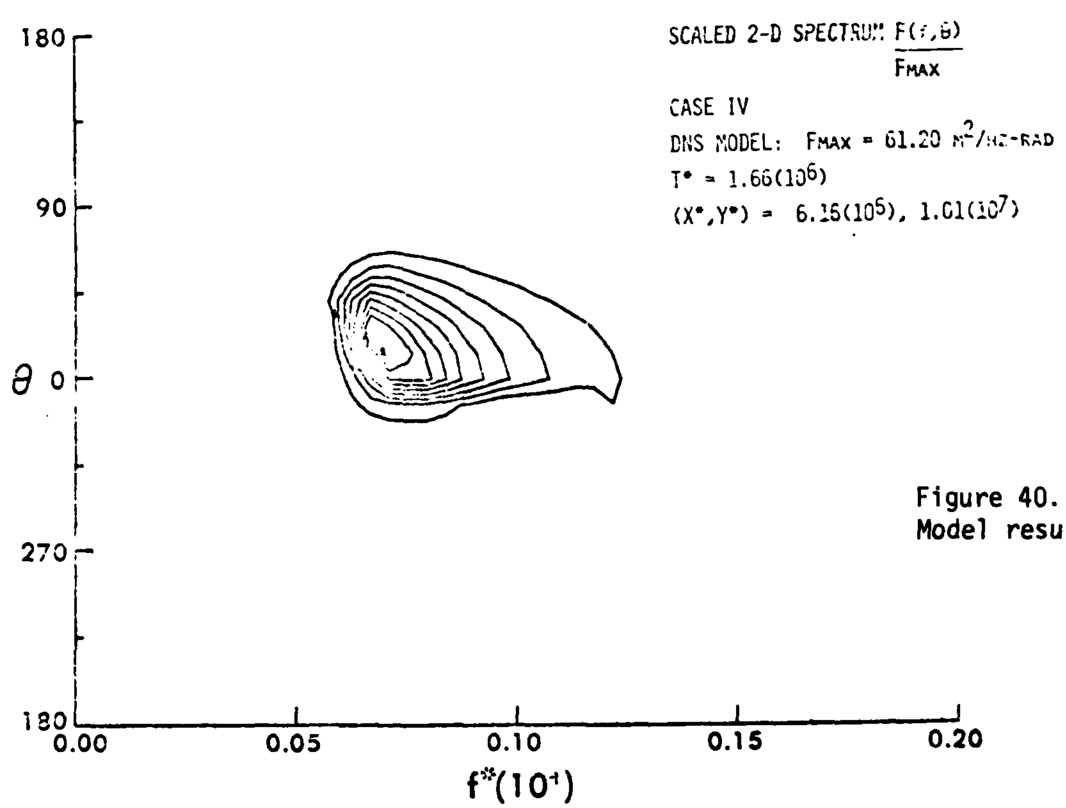


Figure 40.
Model results, Case IV

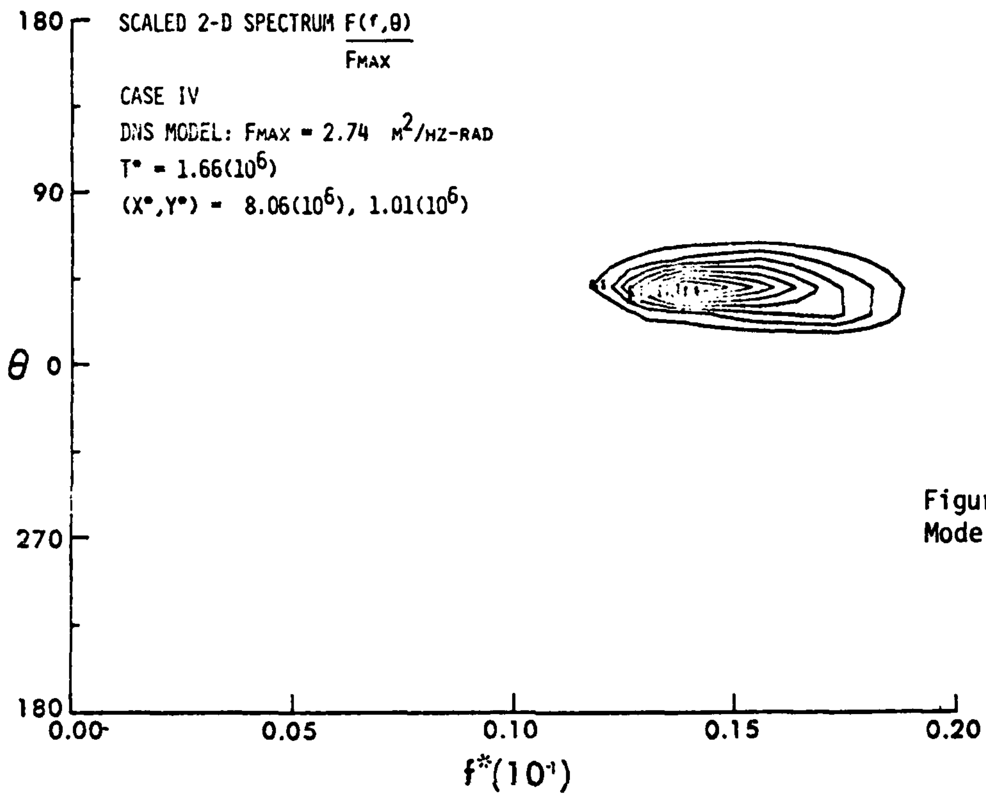


Figure 41.
Model results, Case IV

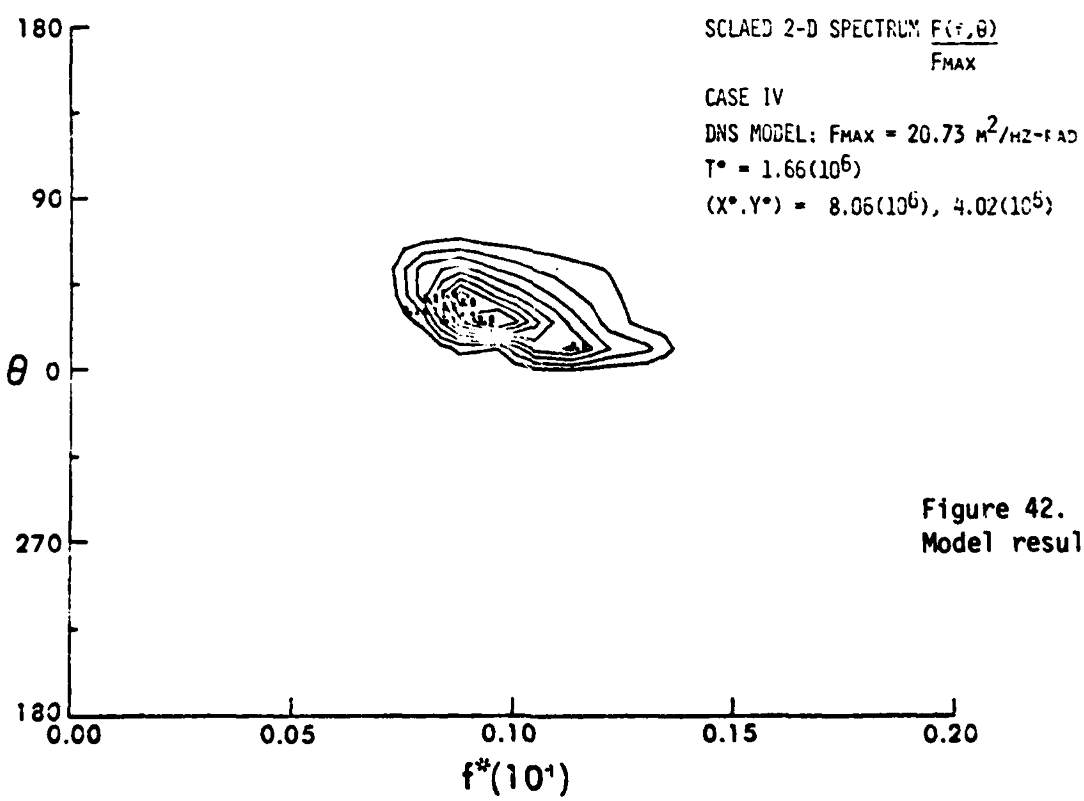


Figure 42.
Model results, Case IV

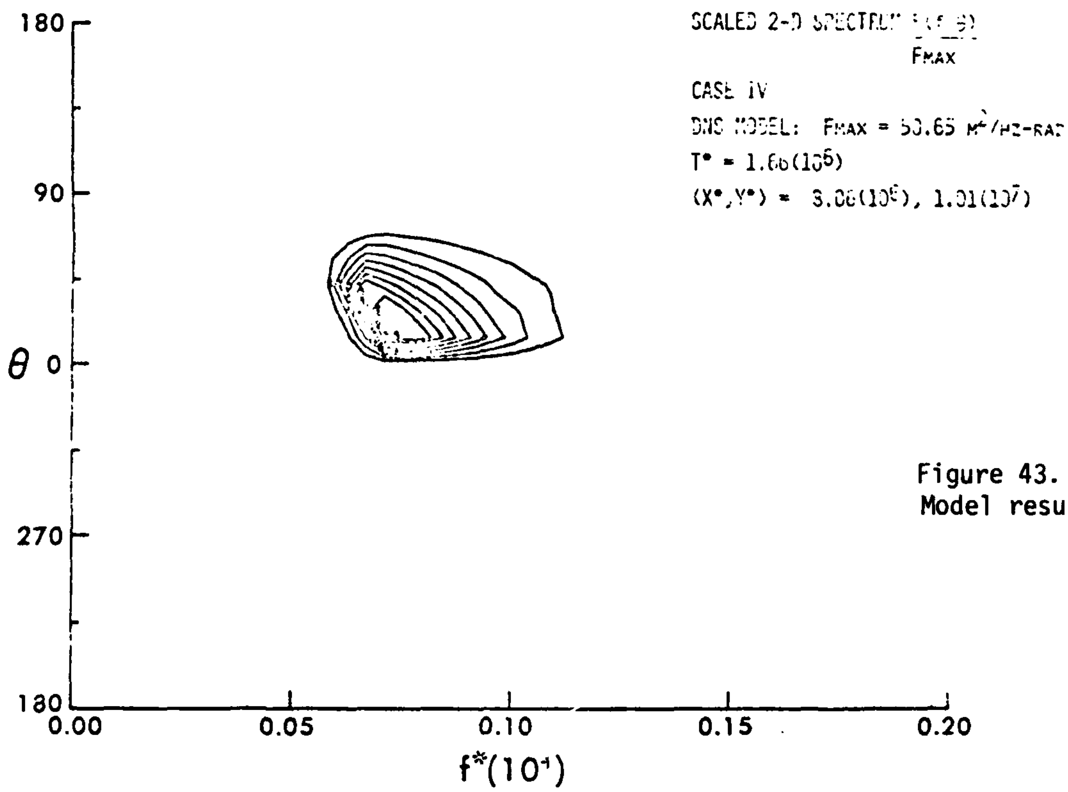


Figure 43.
Model results, Case IV

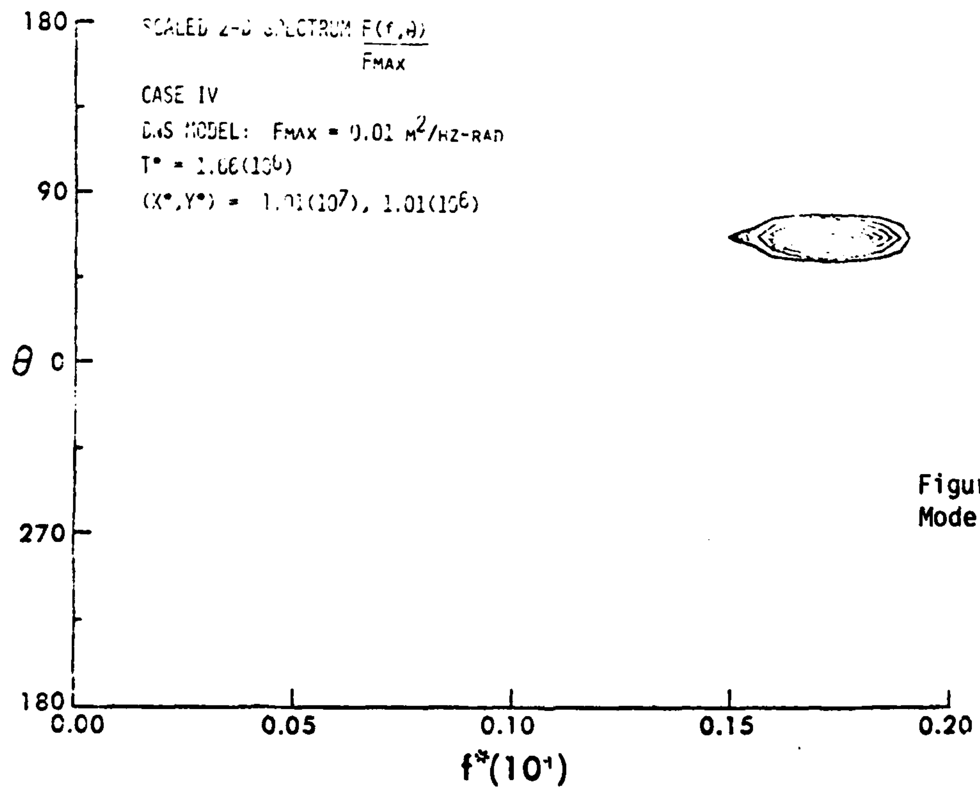


Figure 44.
Model results, Case IV

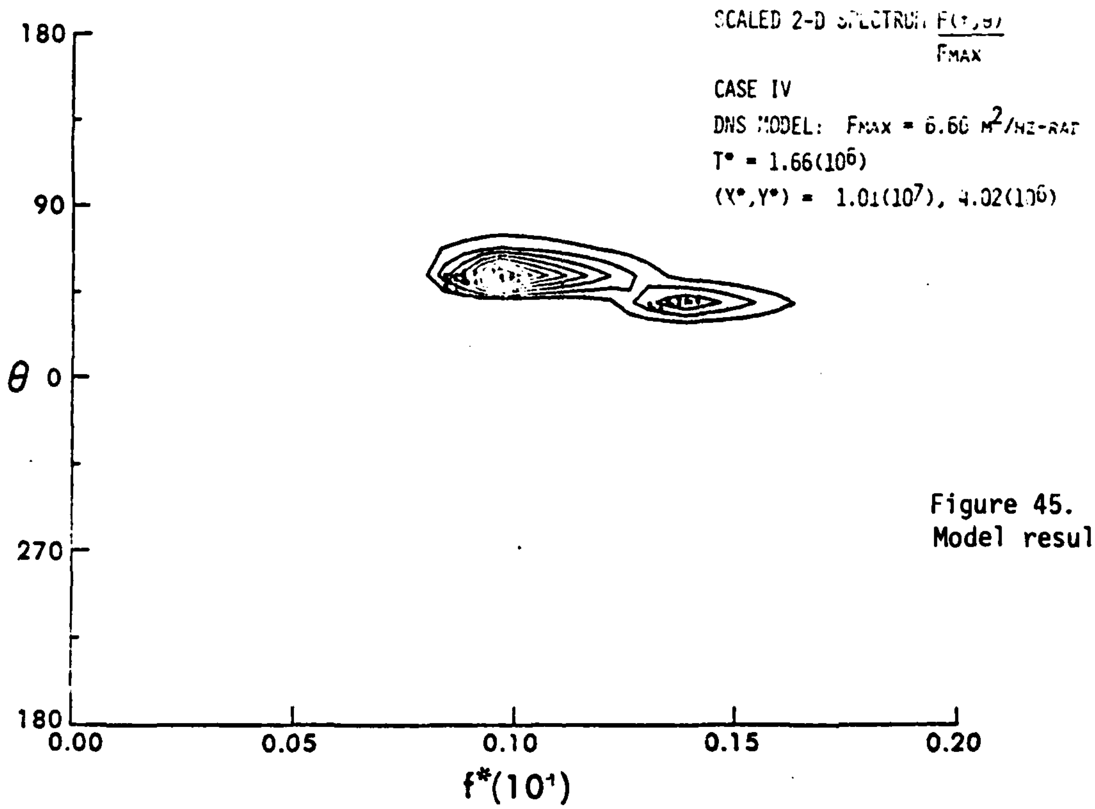


Figure 45.
Model results, Case IV

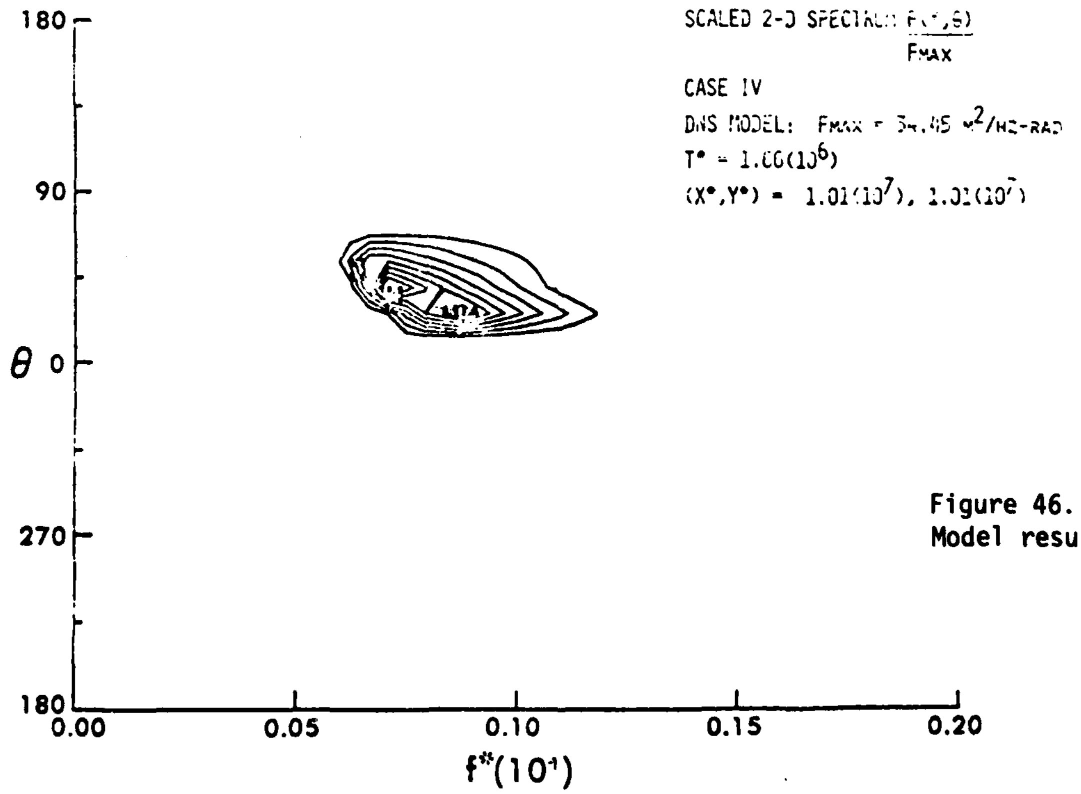


Figure 46.
Model results, Case IV

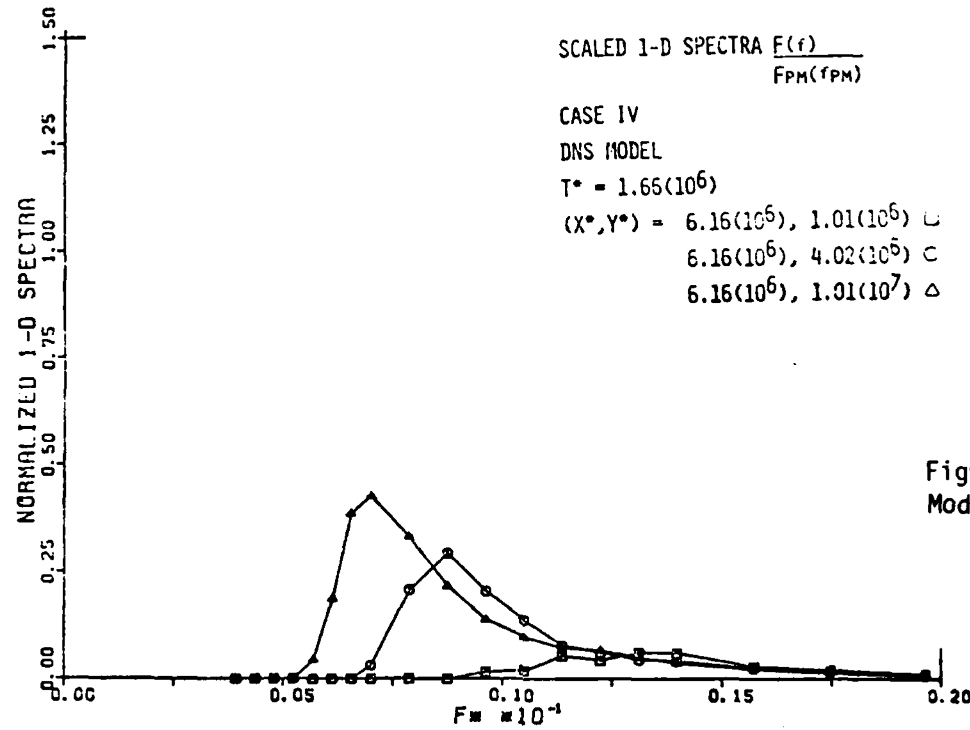


Figure 47.
Model results, Case IV

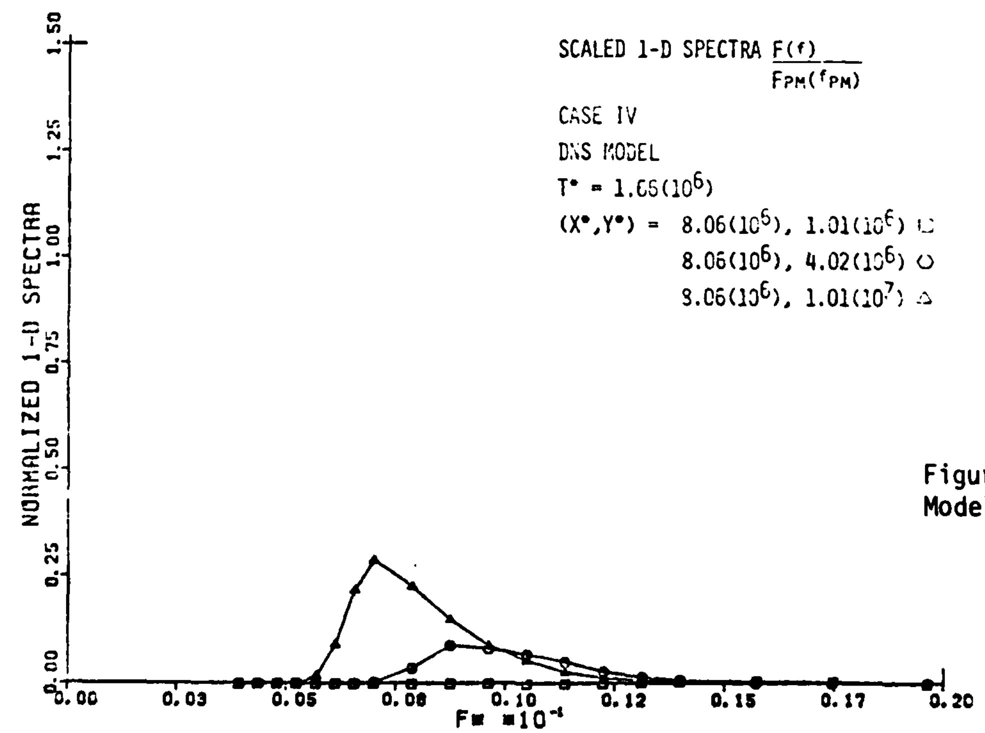


Figure 48.
Model results, Case IV

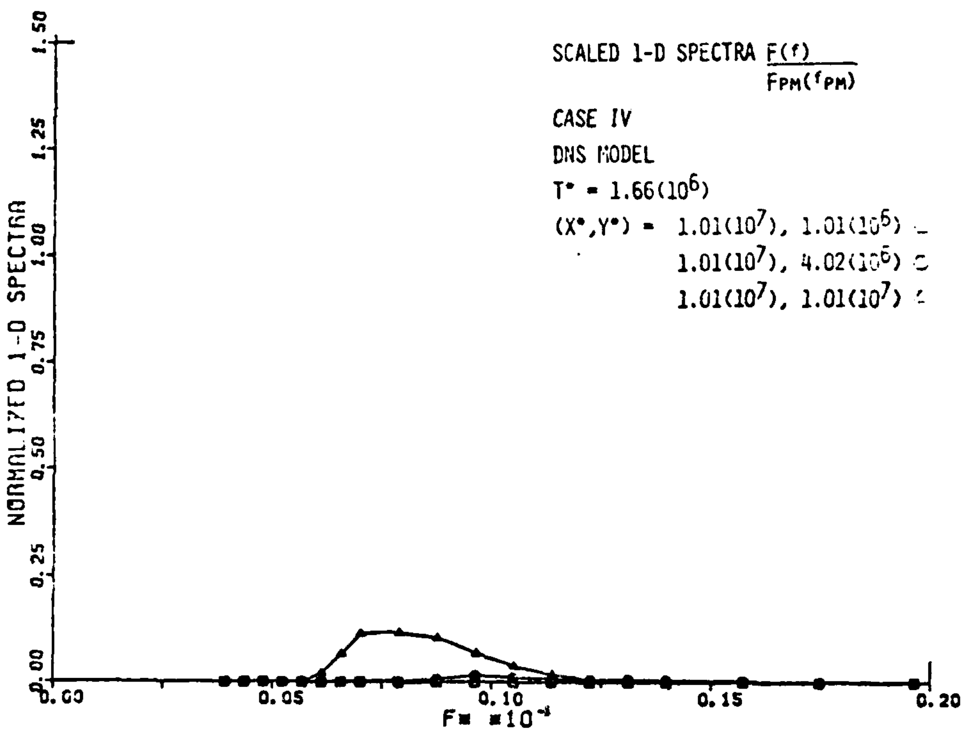


Figure 49. Model results, Case IV

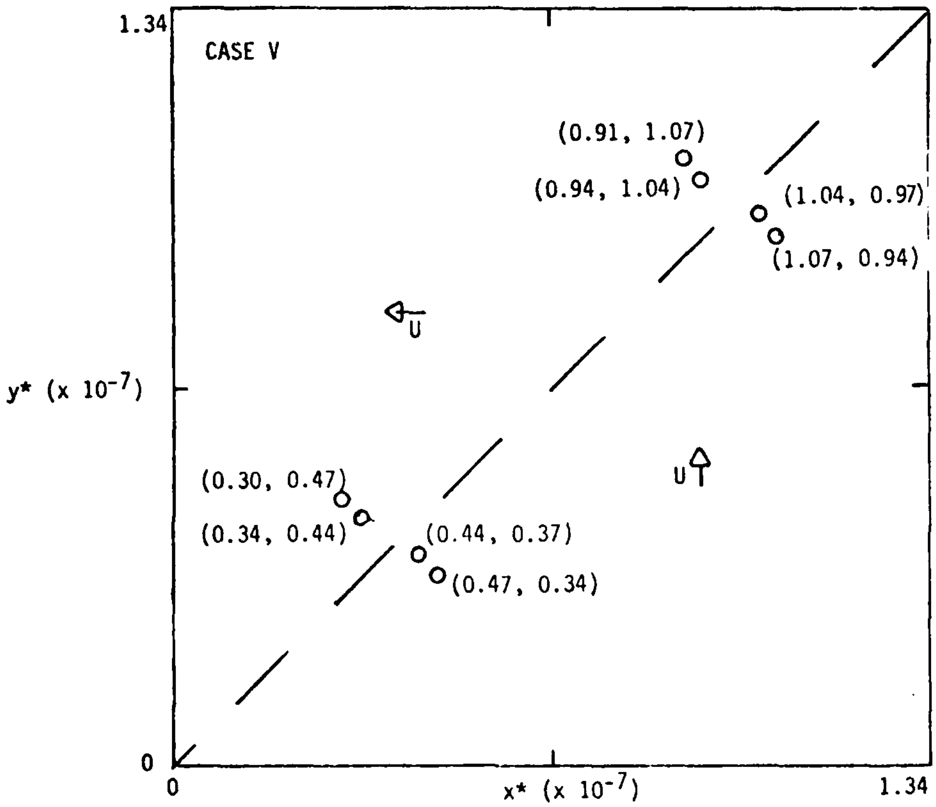


Figure 50. Model output points, Case V

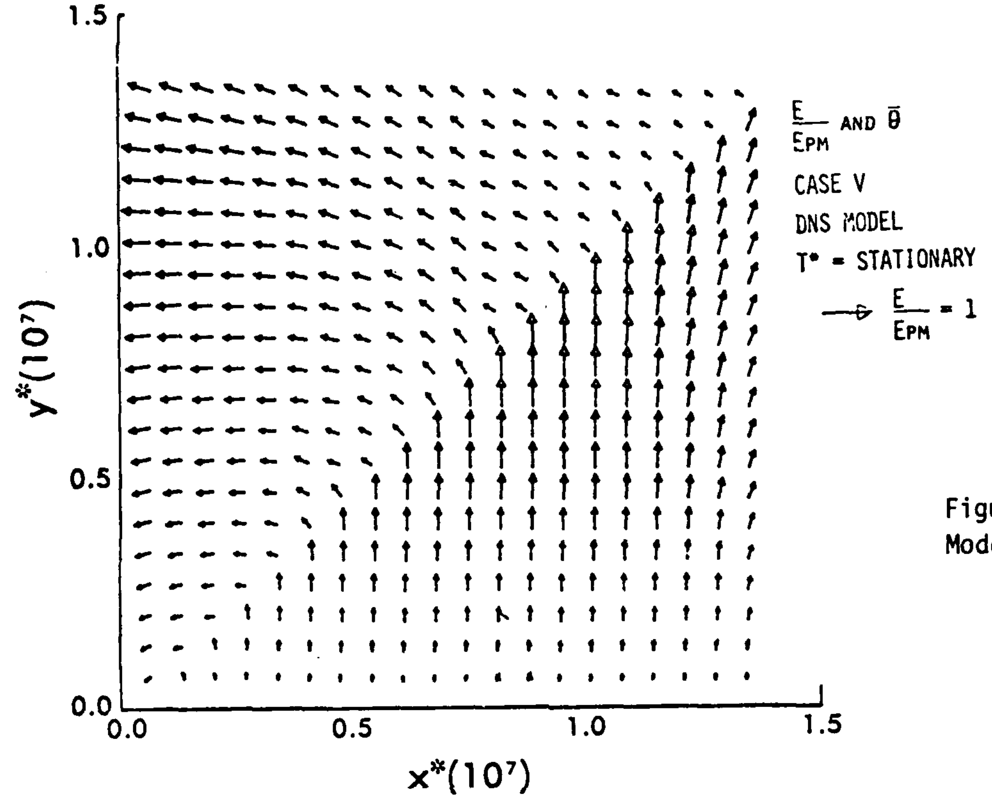


Figure 51.
Model results, Case V

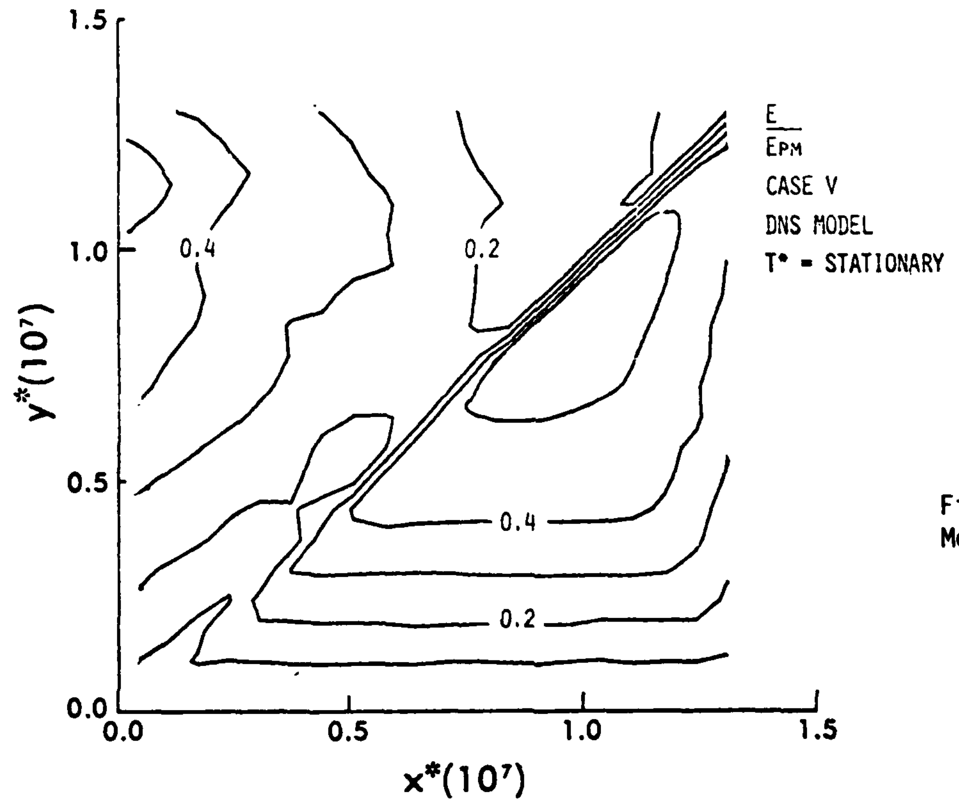


Figure 52.
Model results, Case V

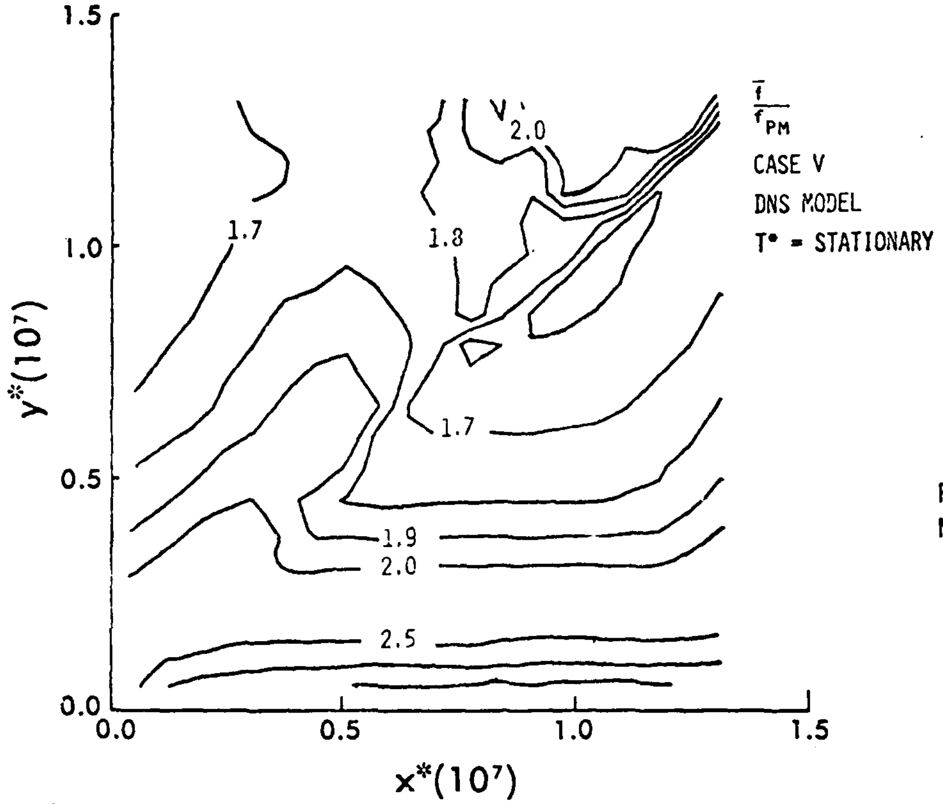


Figure 53.
Model results, Case V

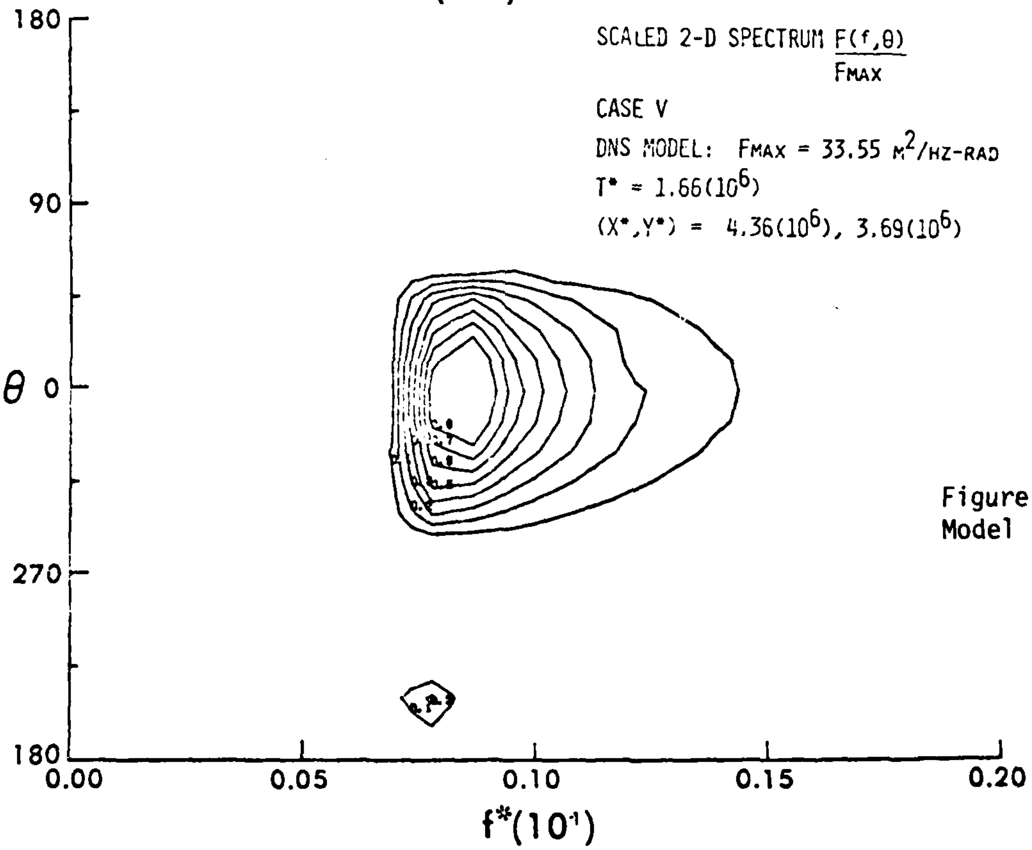


Figure 54.
Model results, Case V

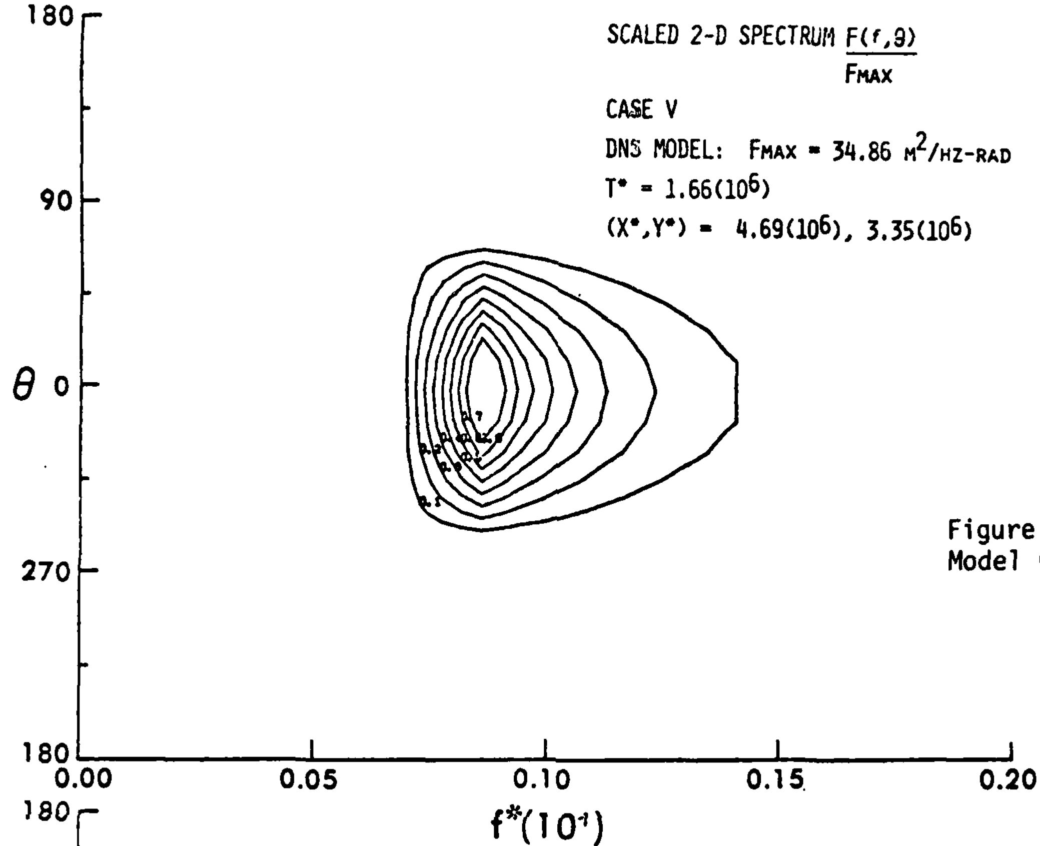


Figure 55.
Model results, Case V

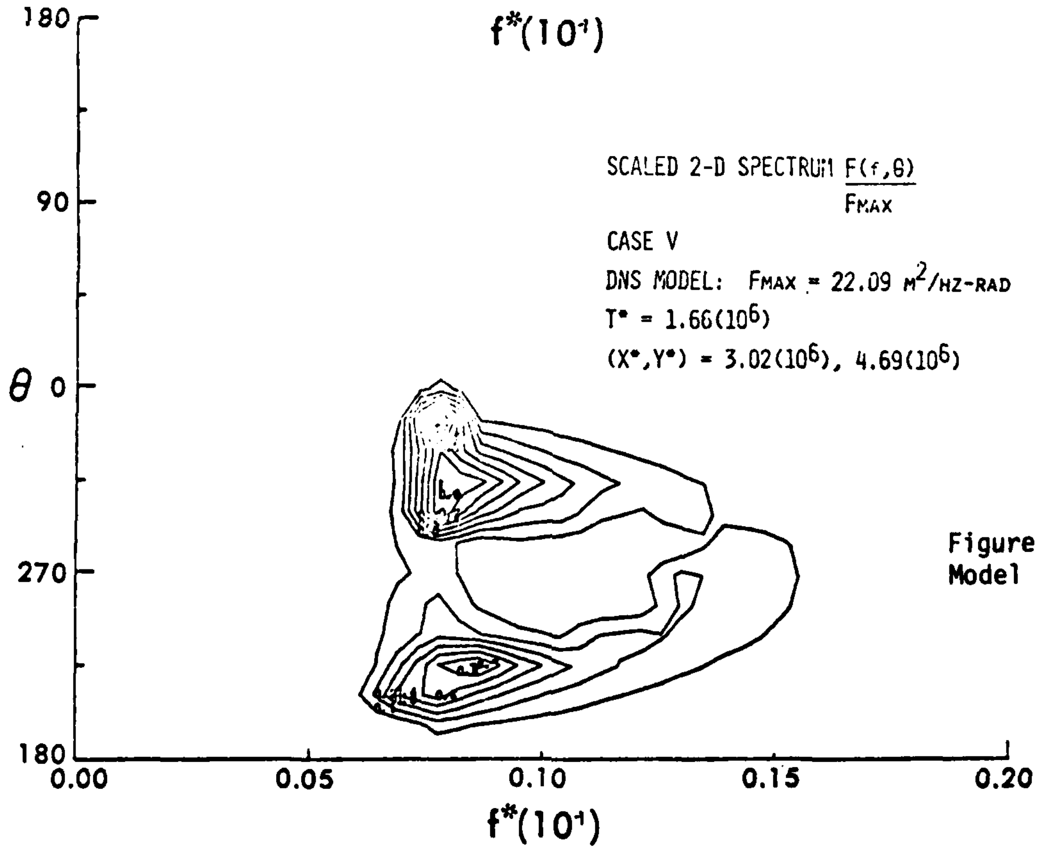


Figure 56.
Model results, Case V

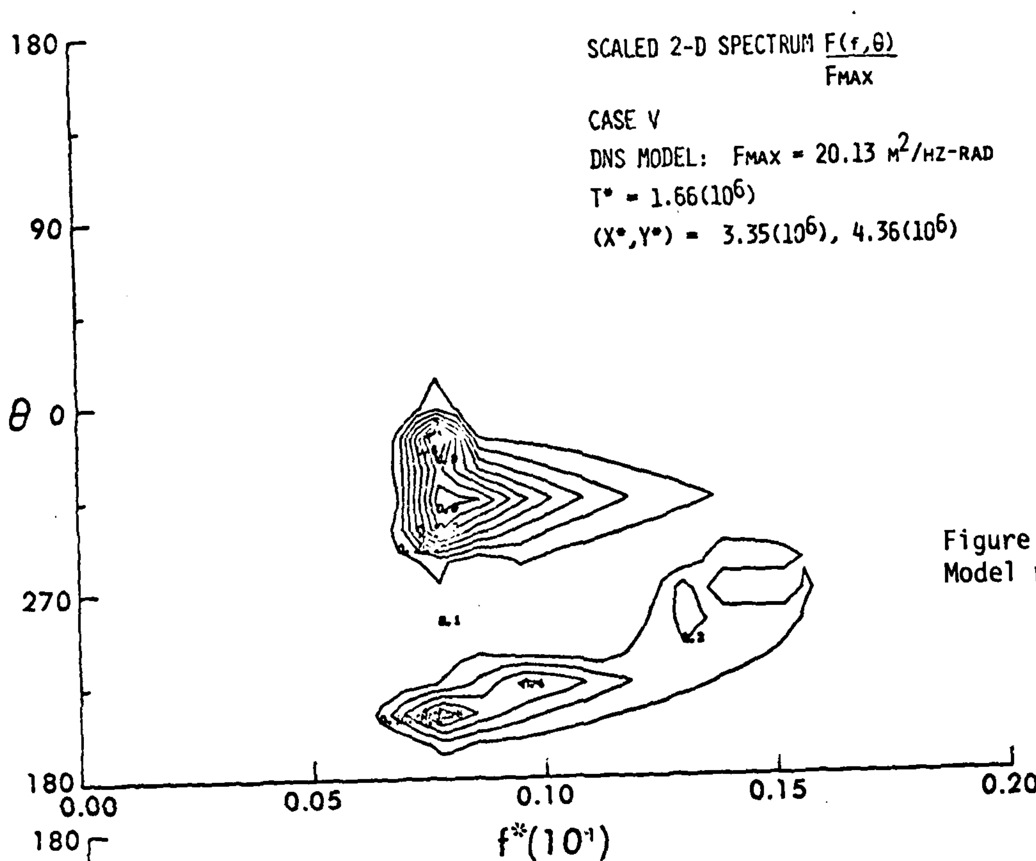


Figure 57.
Model results, Case V

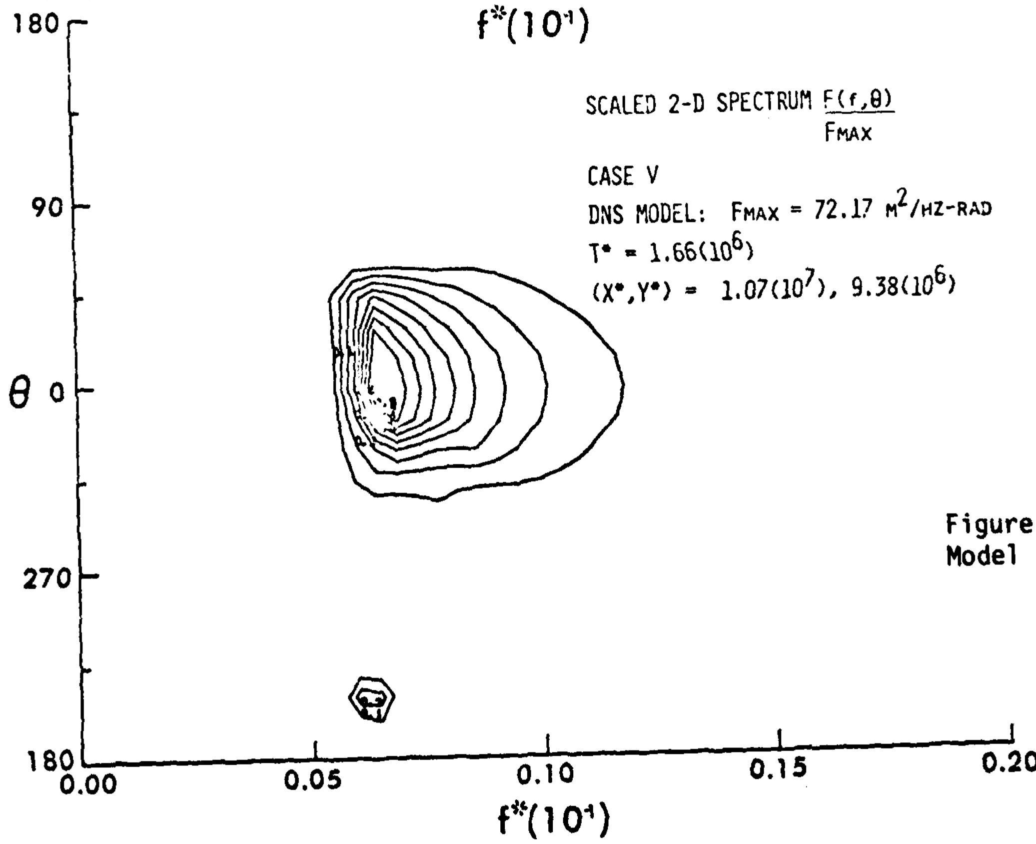


Figure 58.
Model results, Case V

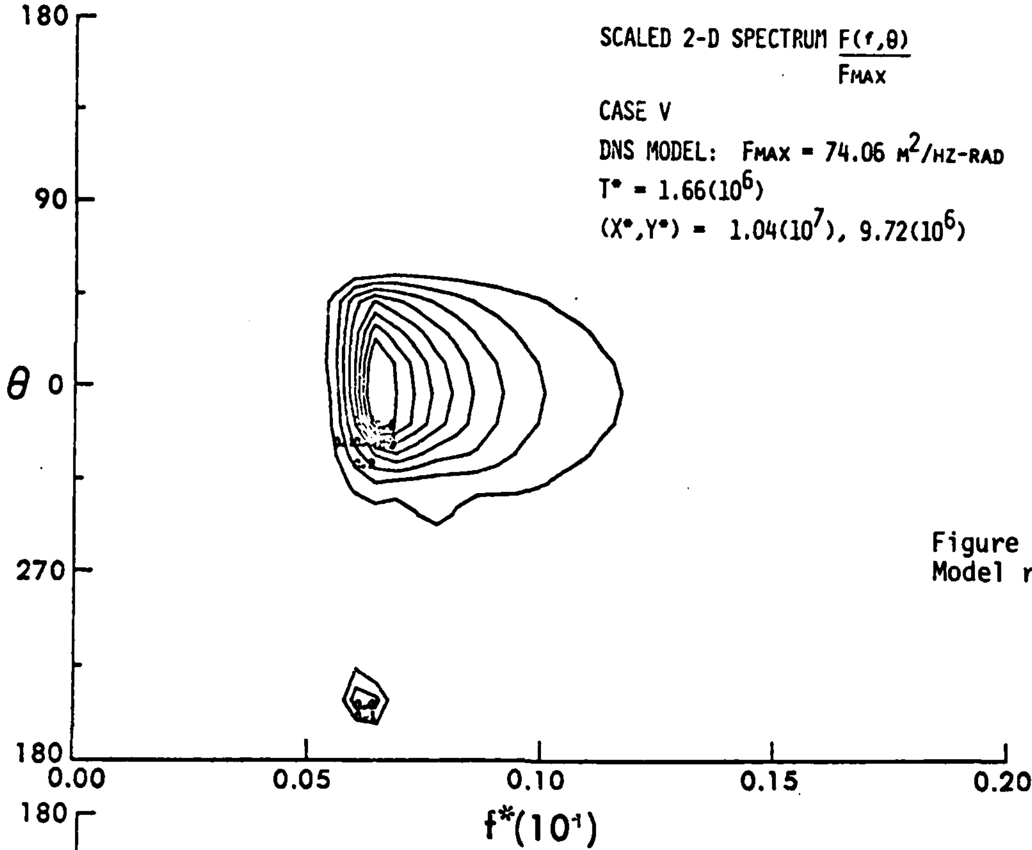


Figure 59.
Model results, Case V

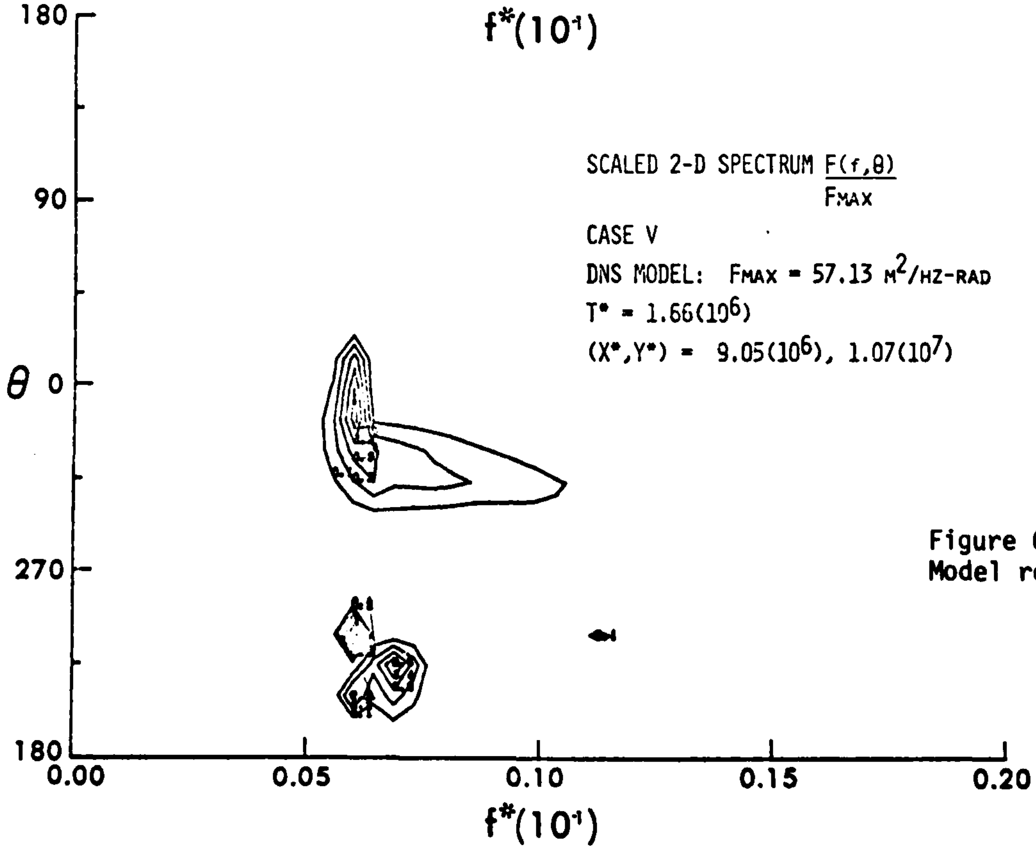


Figure 60.
Model results, Case V

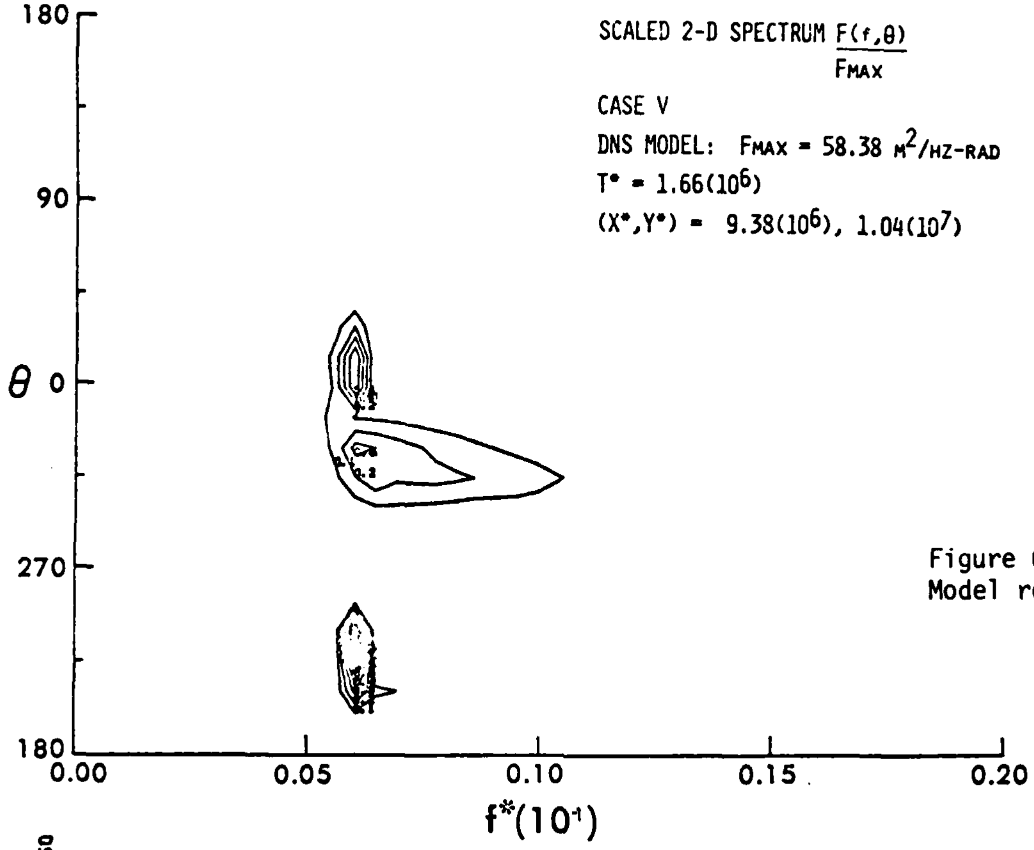


Figure 61.
Model results, Case V

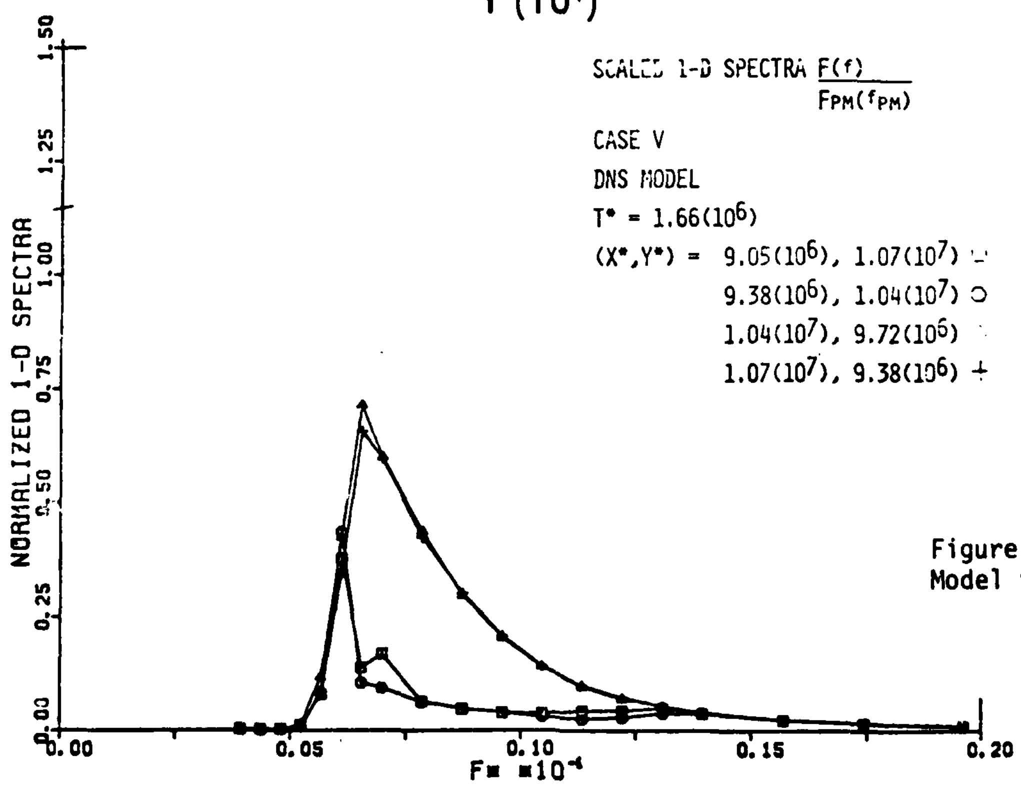


Figure 62.
Model results, Case V

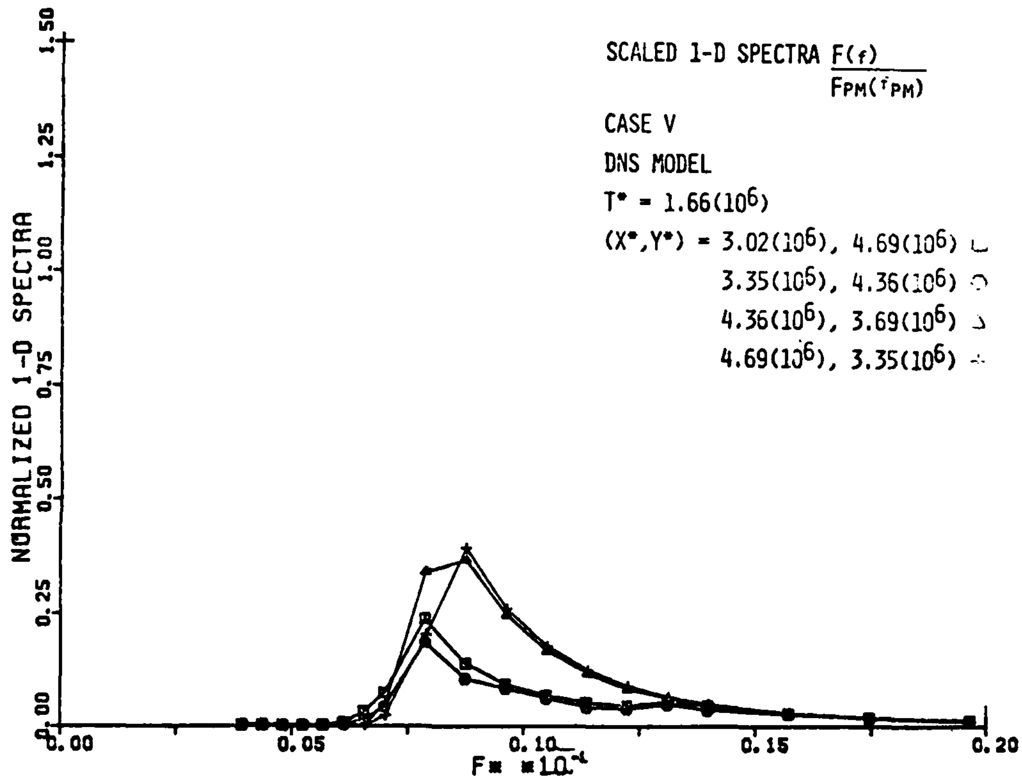


Figure 63. Model results, Case V

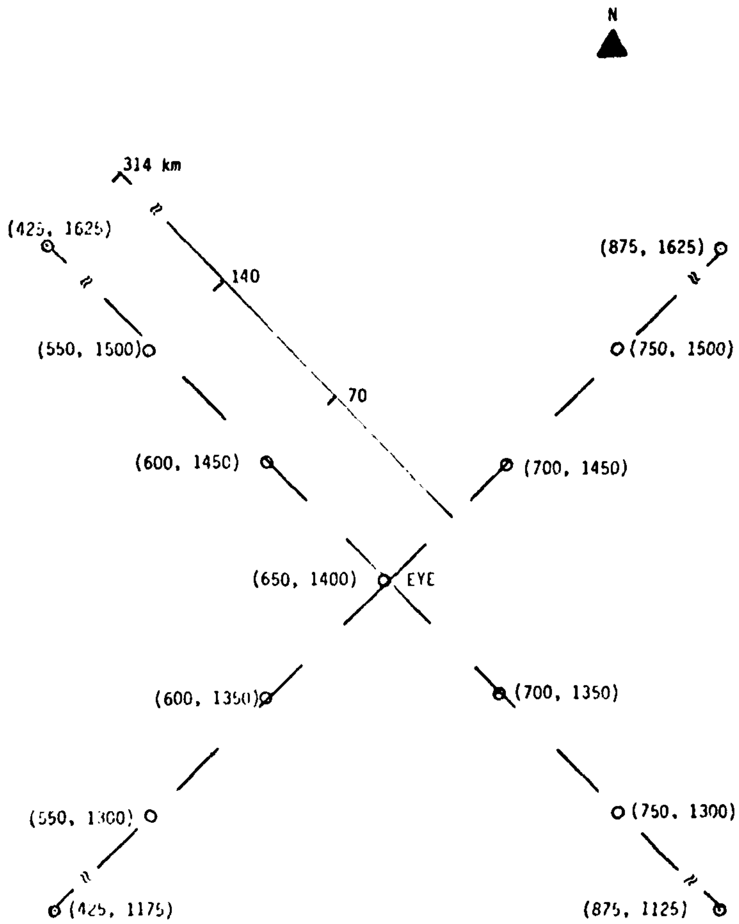


Figure 64. Model output points, Cases VI.A-B

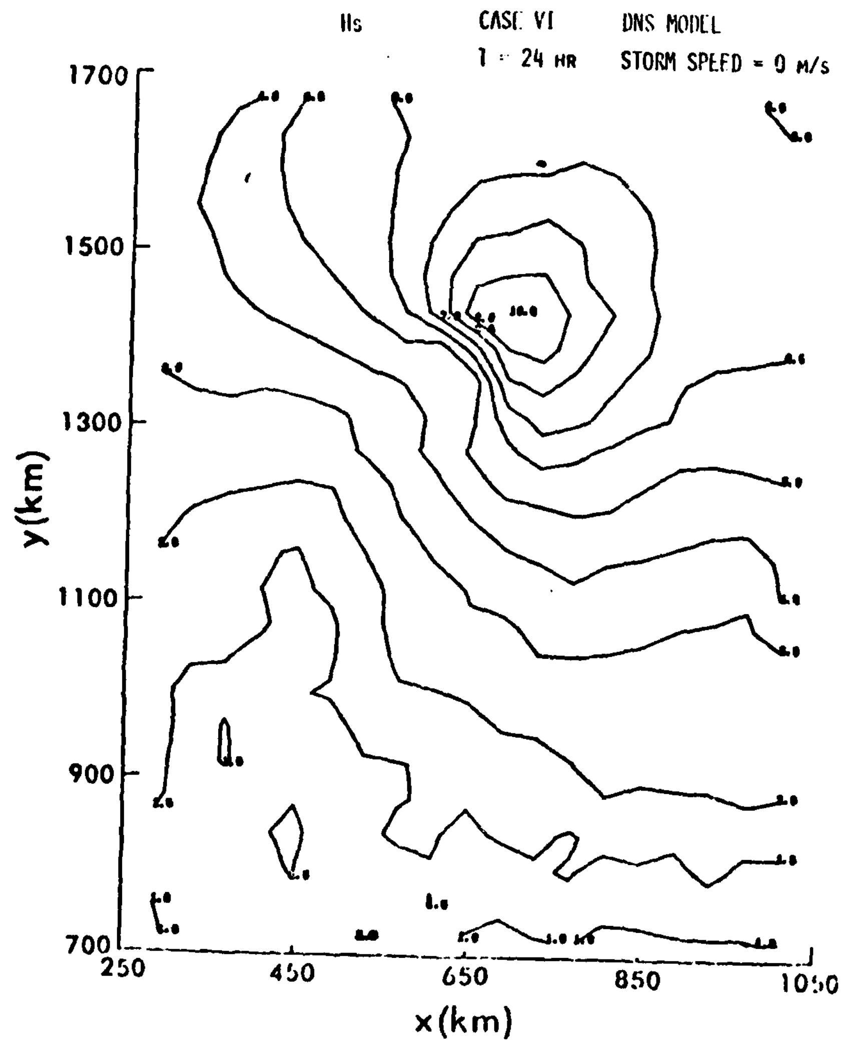


Figure 65. Model results, Case VI.A

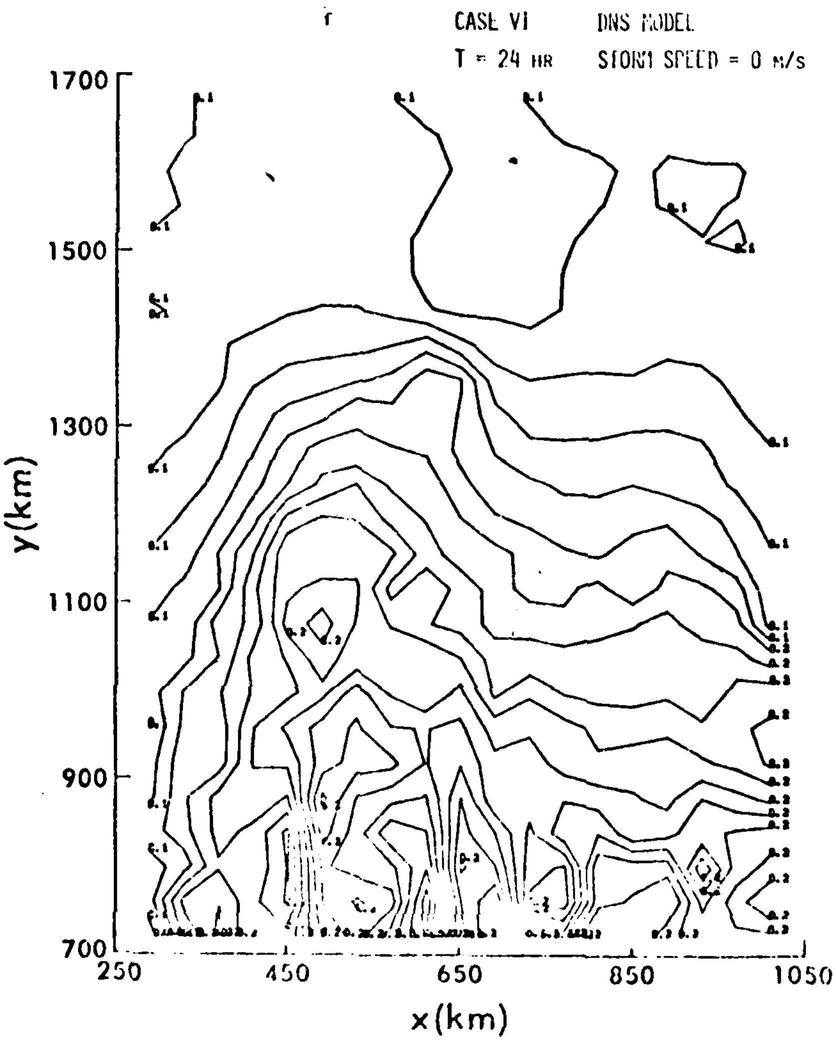


Figure 66. Model results, Case VI.A

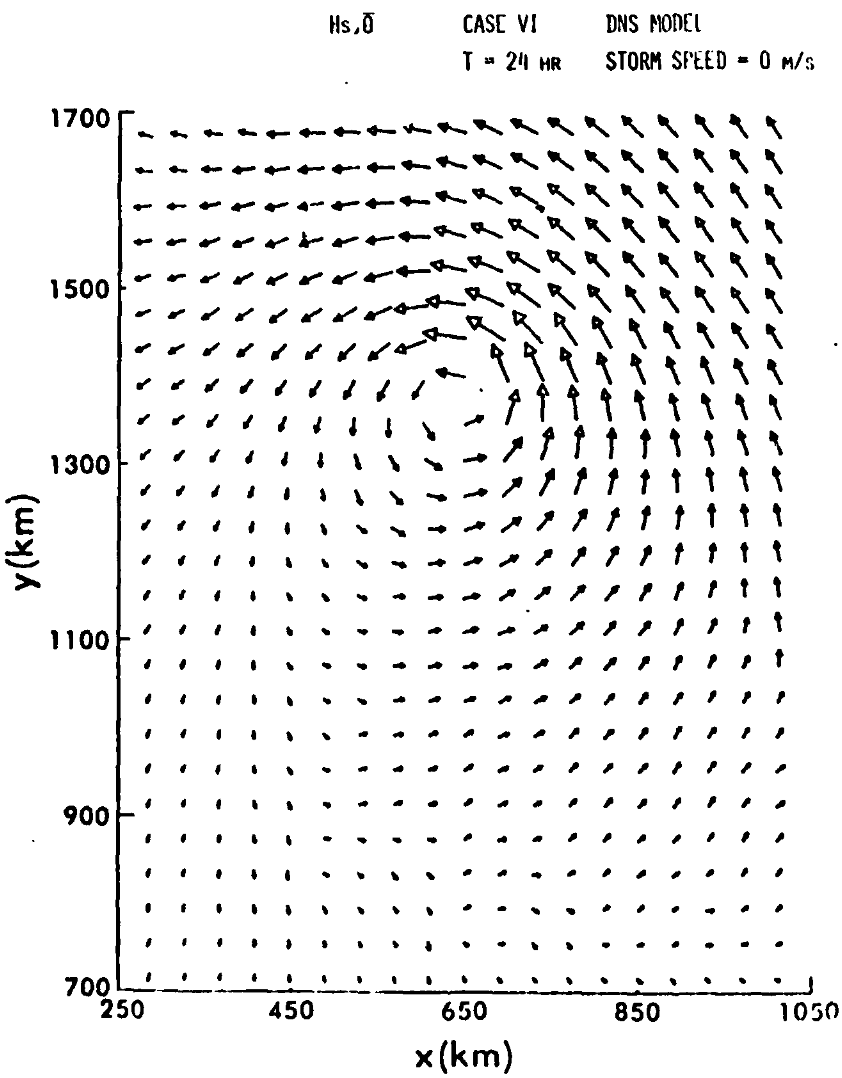


Figure 67. Model results, Case VI.A

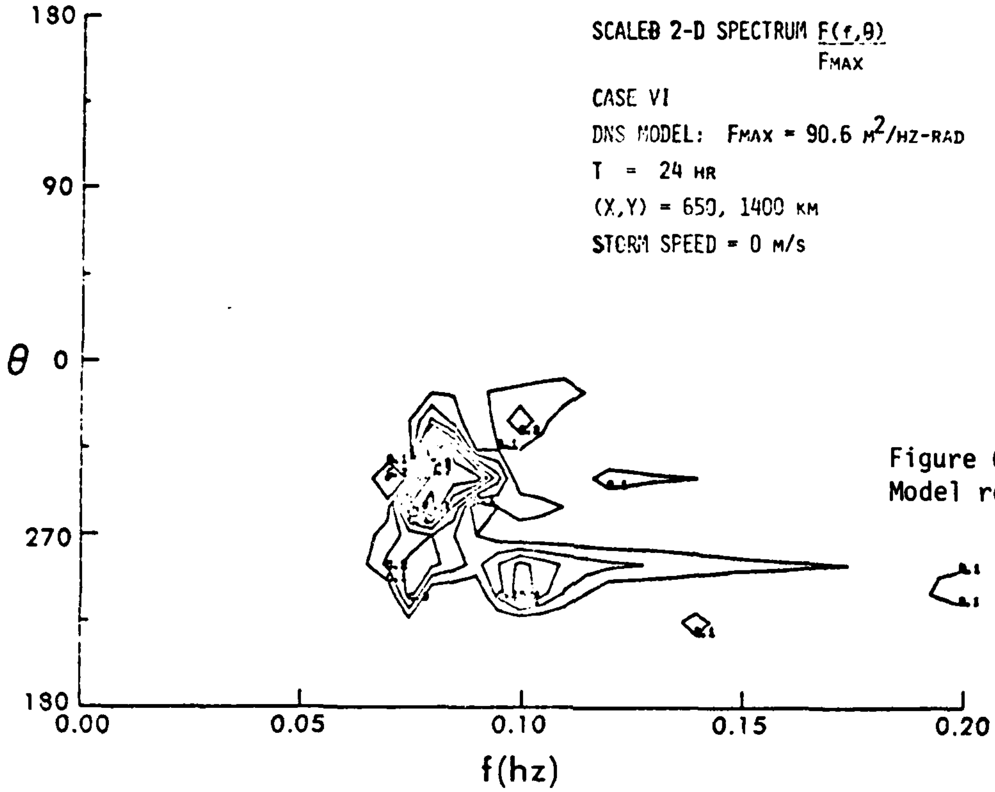


Figure 68.
Model results, Case VI.A

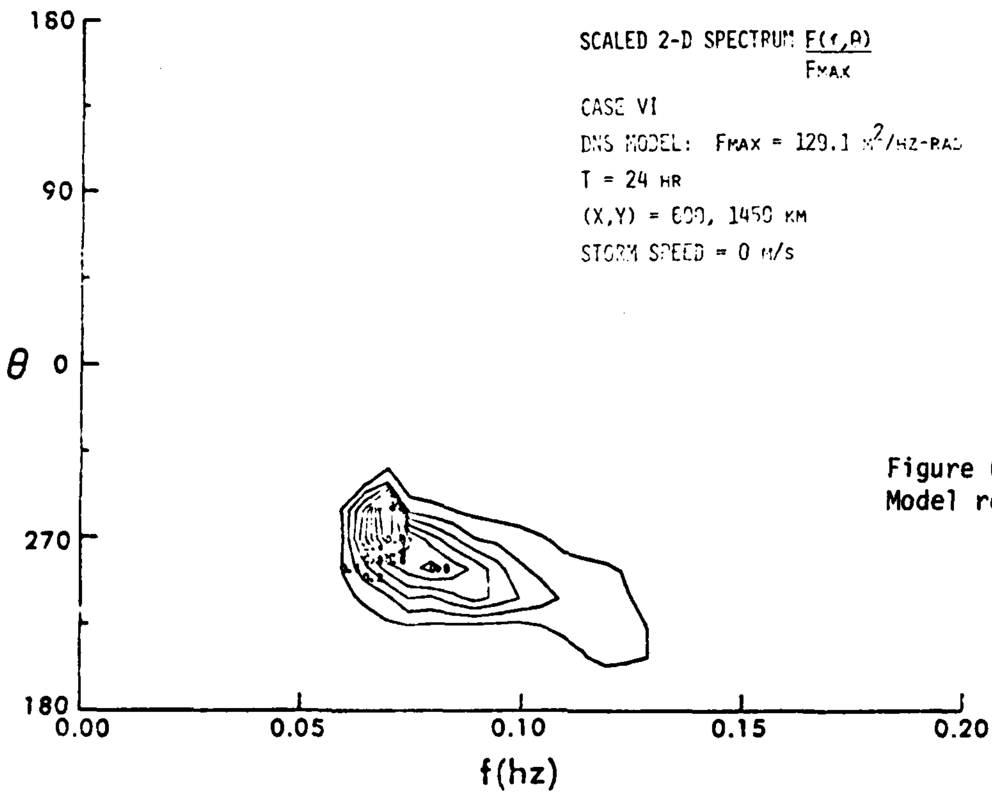


Figure 69.
Model results, Case VI.A

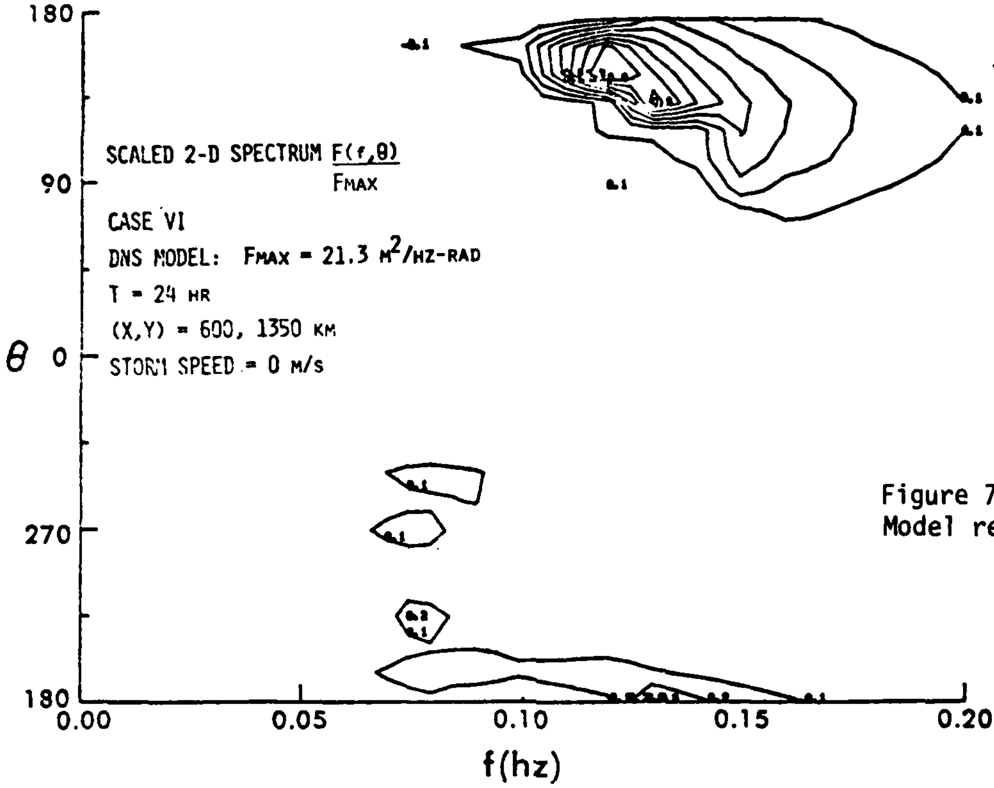


Figure 70.
Model results, Case VI.A

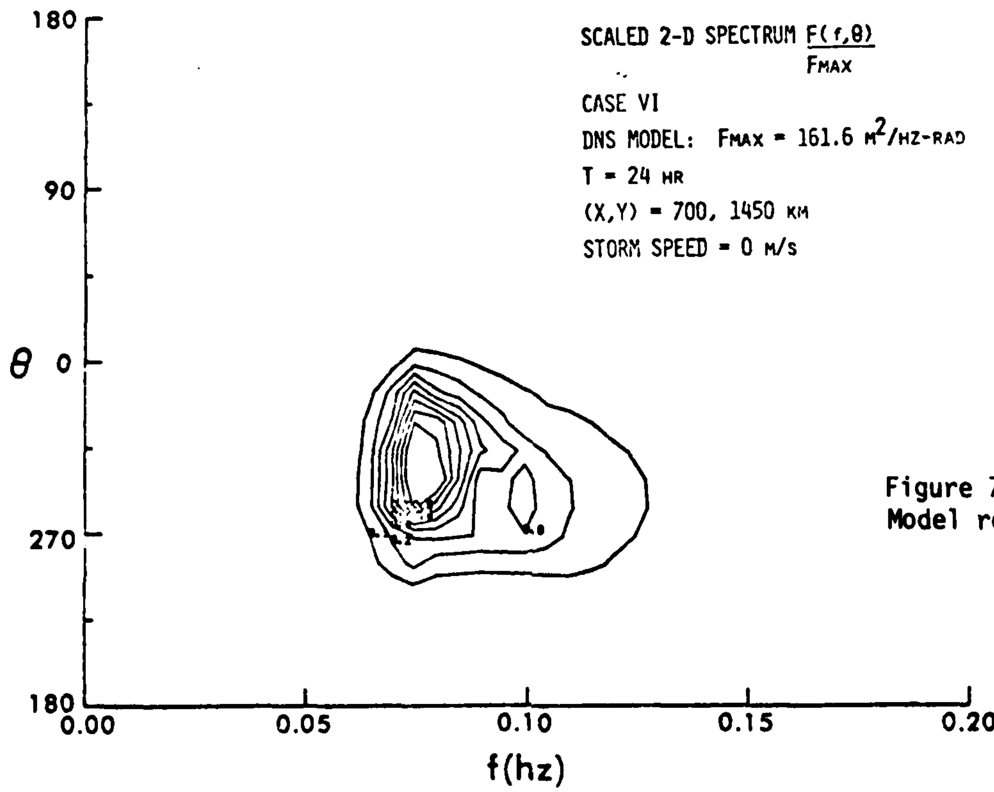


Figure 71.
Model results, Case VI.A

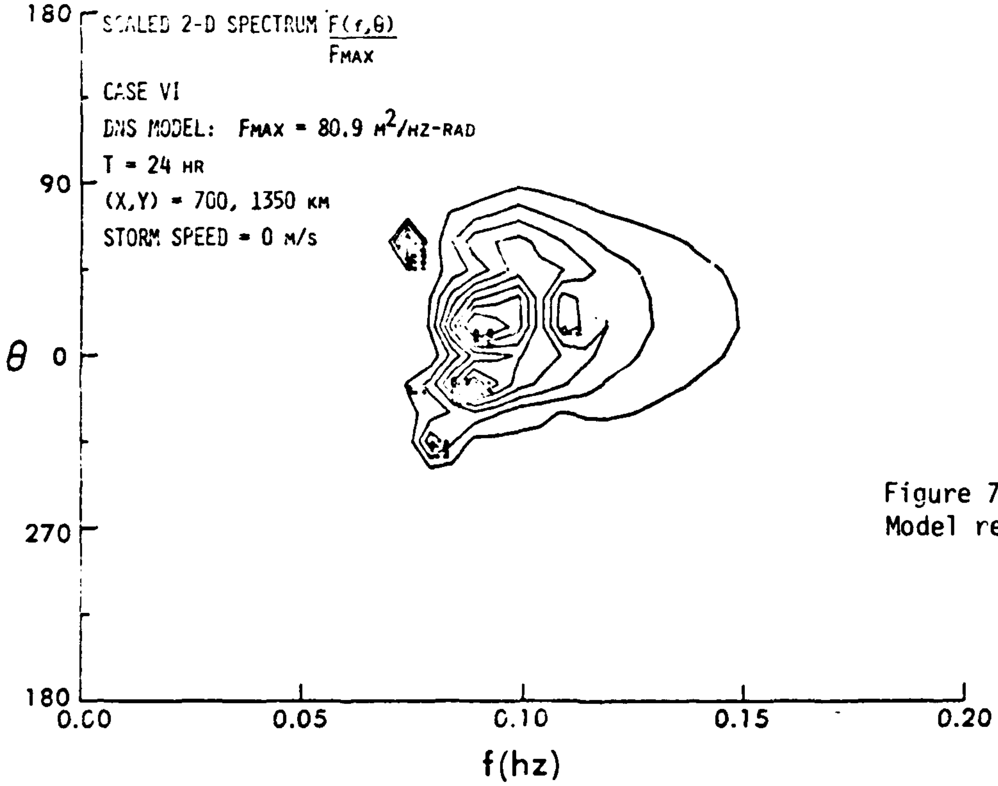


Figure 72.
Model results, Case VI.A

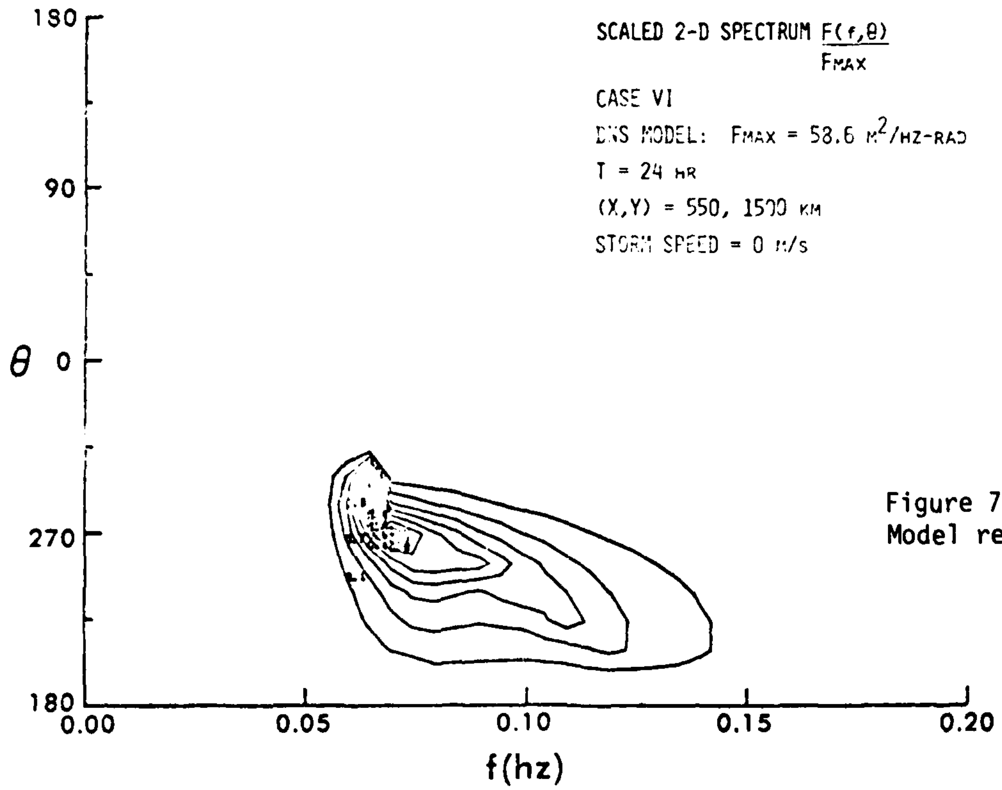


Figure 73.
Model results, Case VI.A

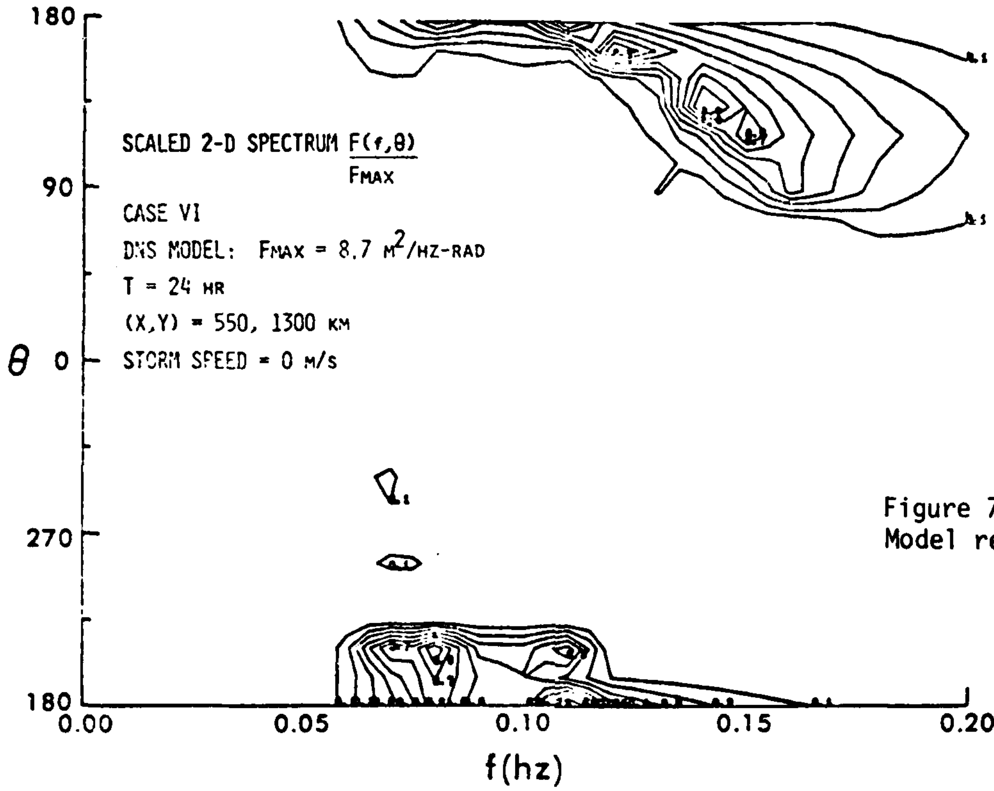


Figure 74.
Model results, Case VI.A

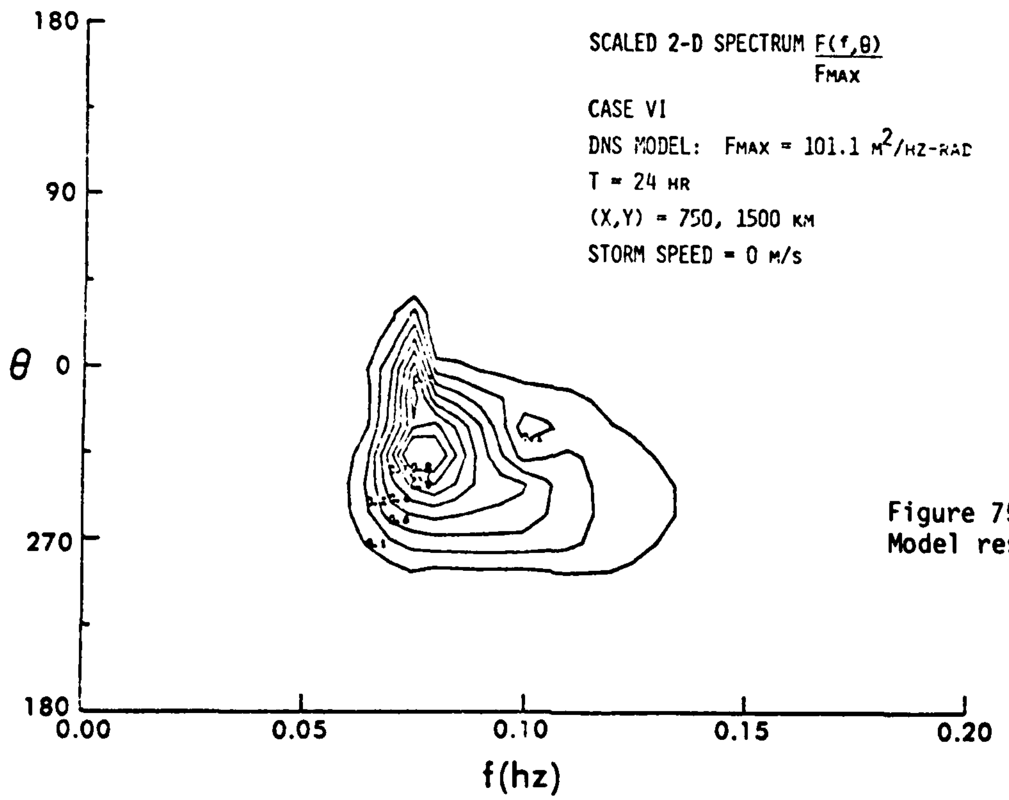


Figure 75.
Model results, Case VI.A

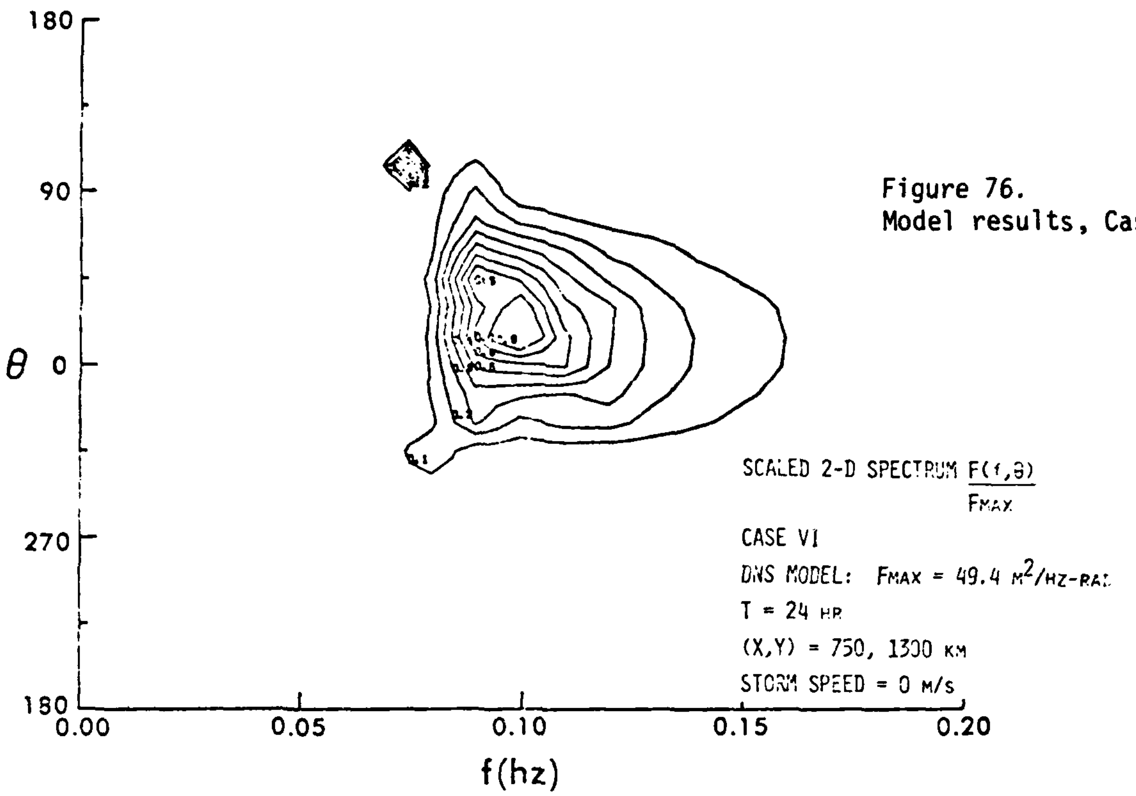


Figure 76.
Model results, Case VI.A

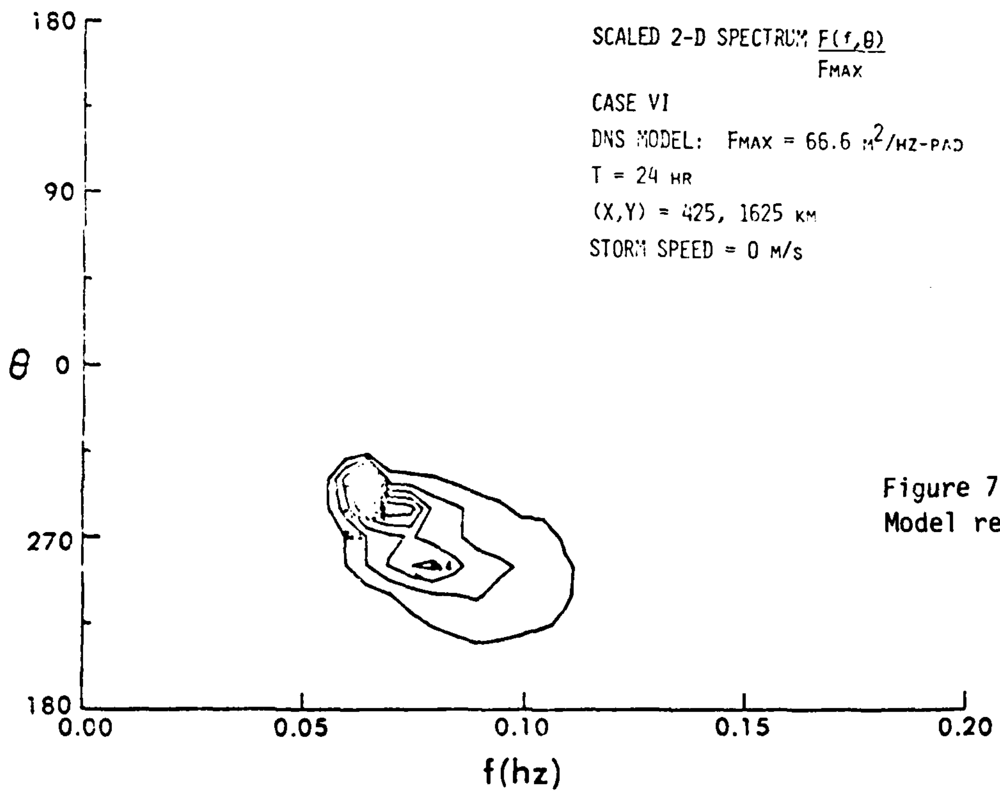


Figure 77.
Model results, Case VI.A

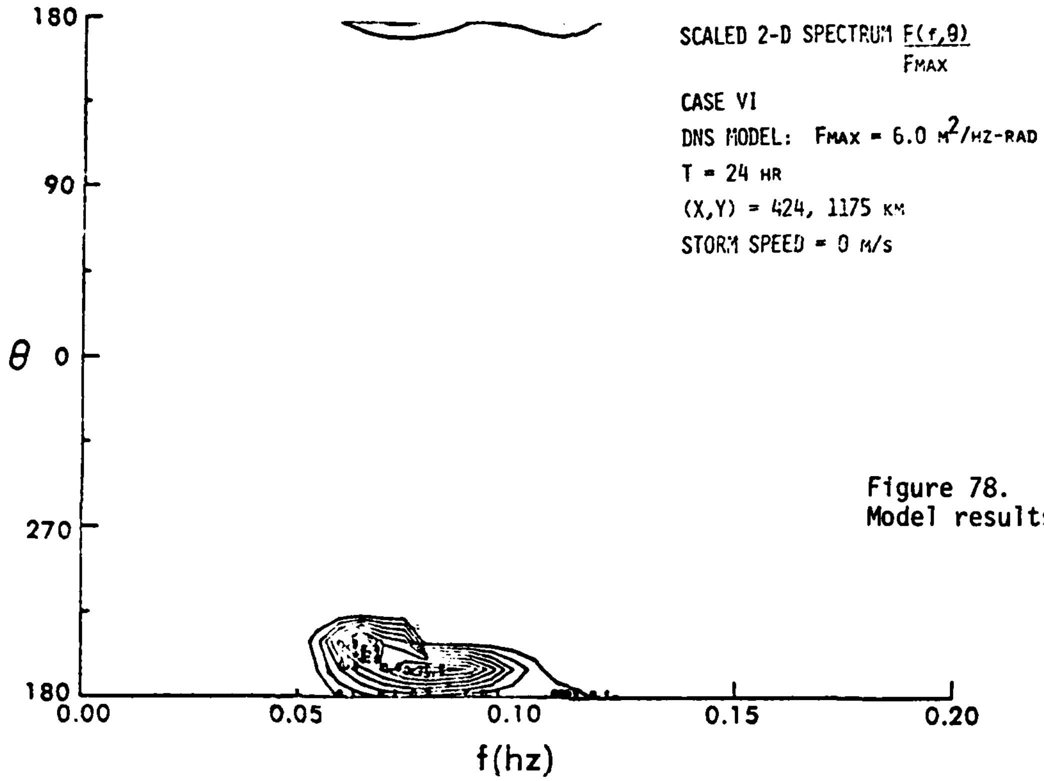


Figure 78.
Model results, Case VI.A

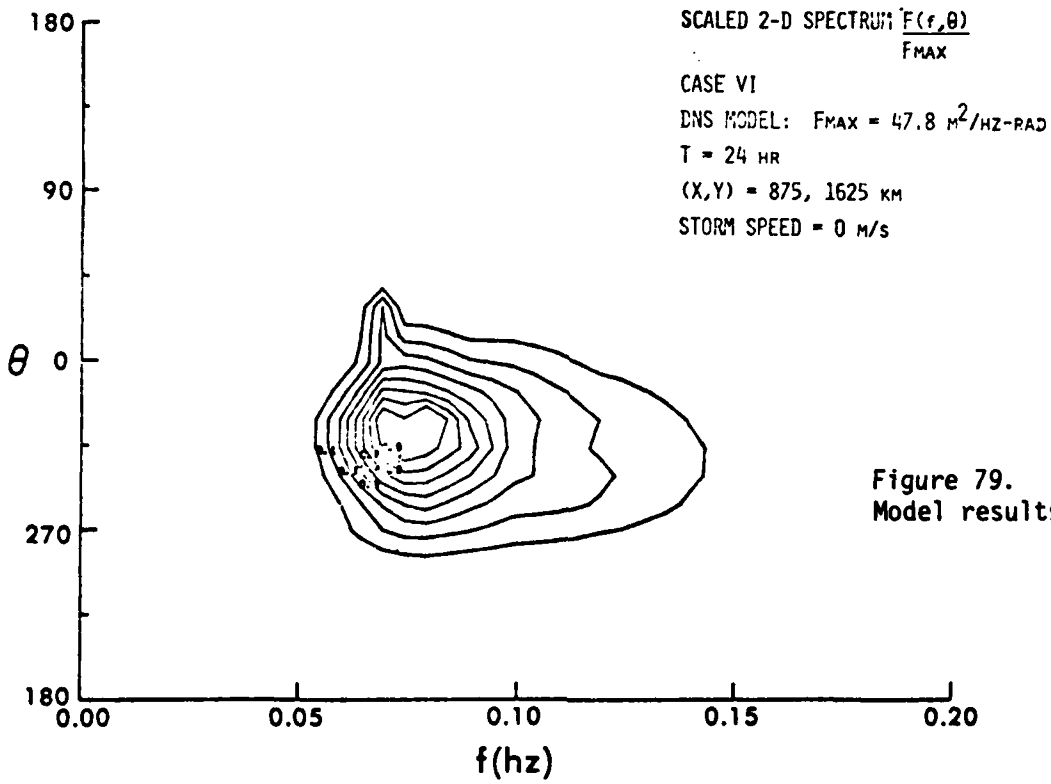


Figure 79.
Model results, Case VI.A

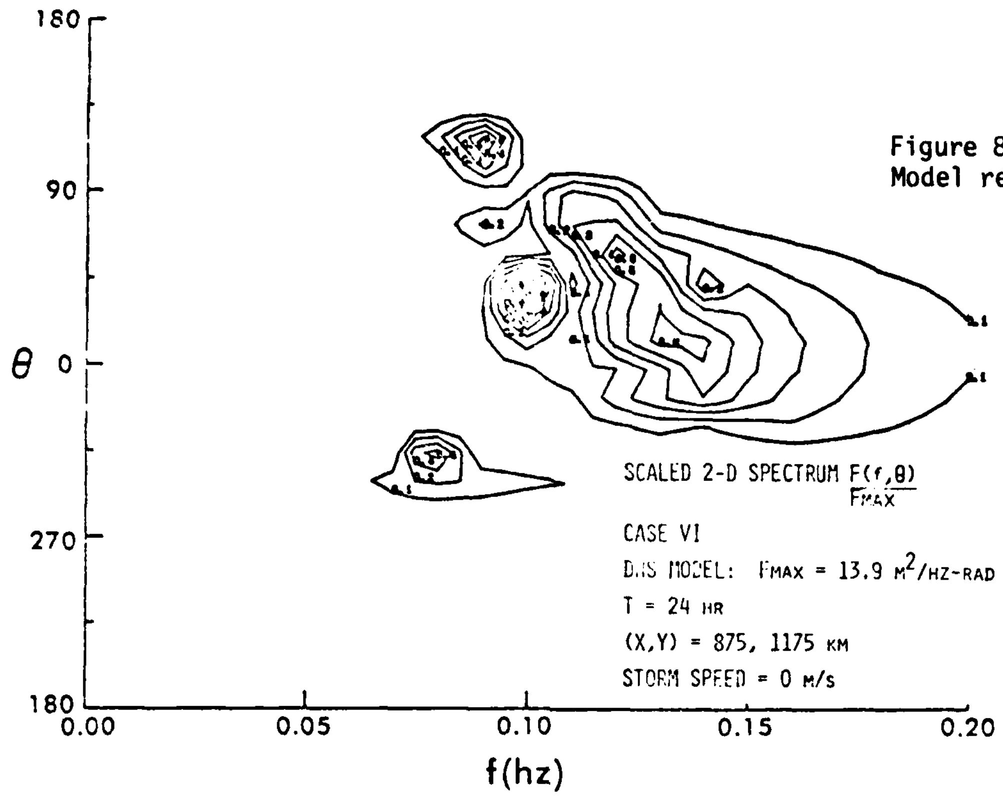


Figure 80.
Model results, Case VI.A

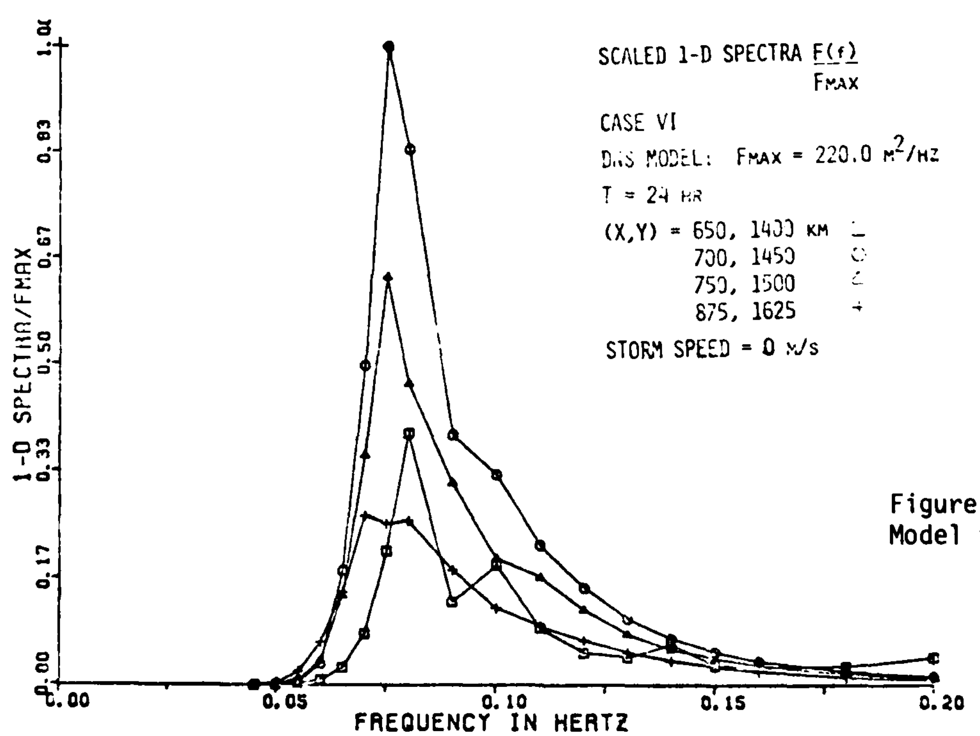


Figure 81.
Model results, Case VI.A

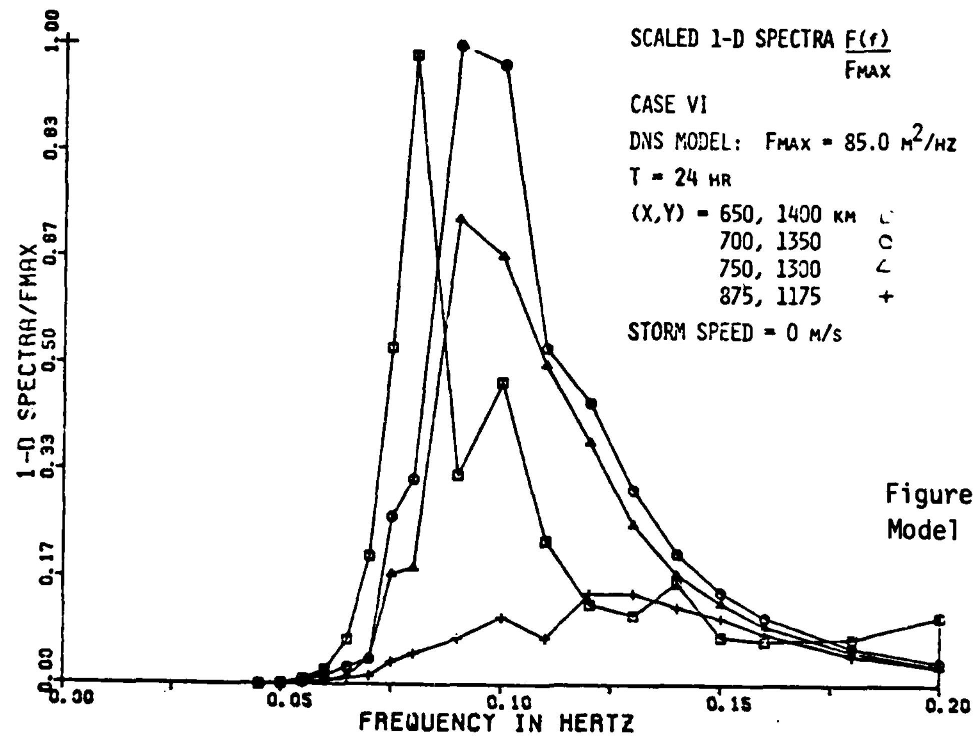


Figure 82.
Model results, Case VI.A

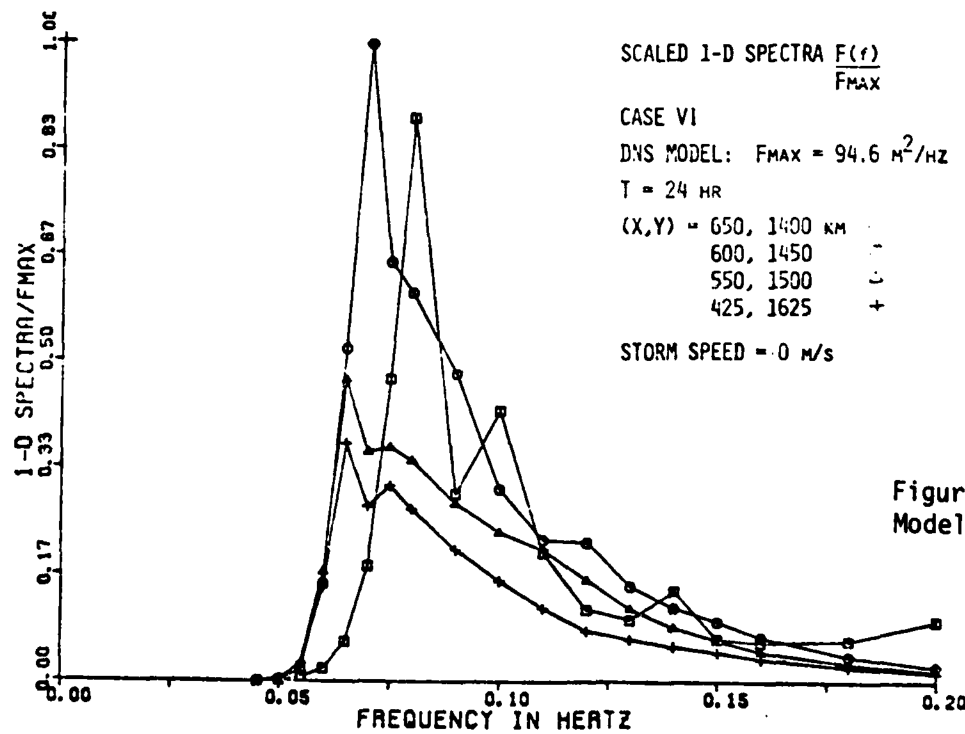


Figure 83.
Model results, Case VI.A

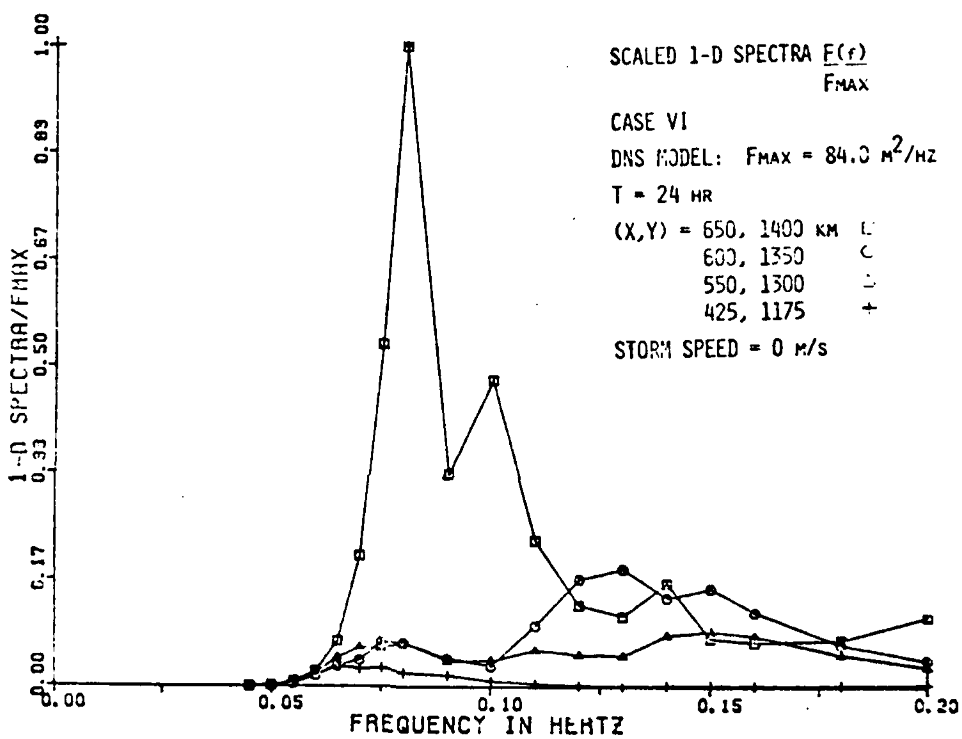


Figure 84. Model results, Case VI.A

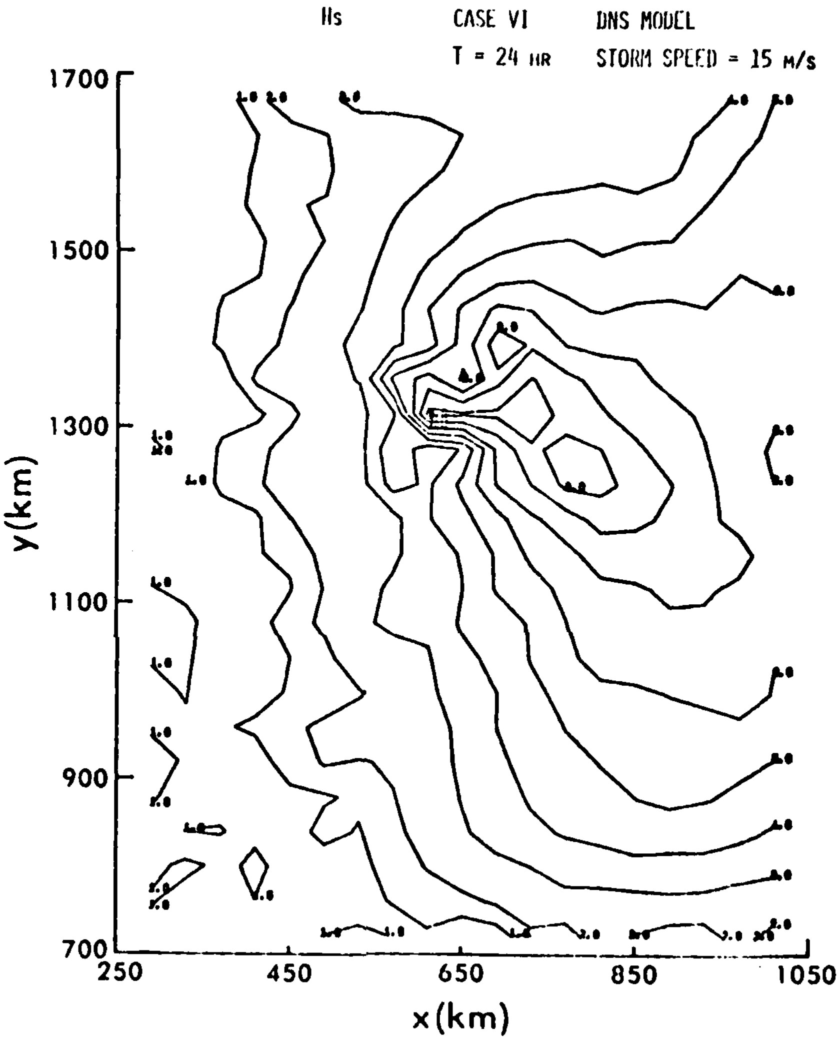


Figure 85. Model results, Case VI.B

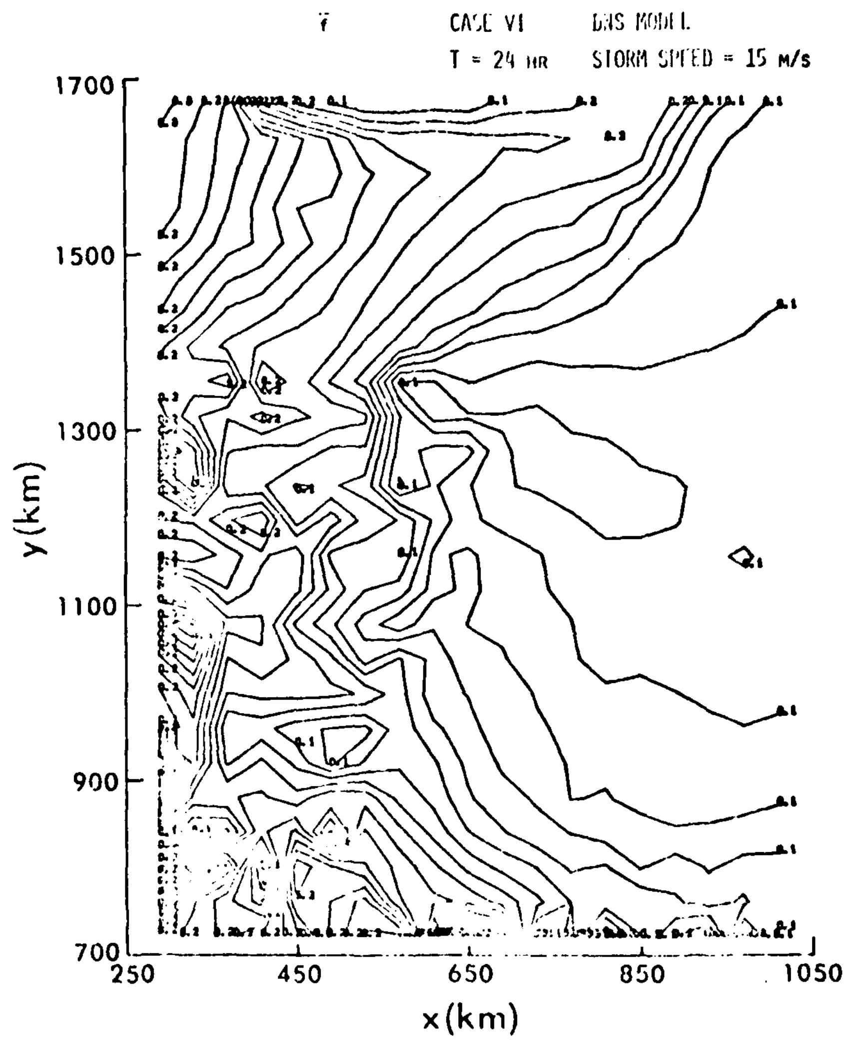


Figure 86. Model results, Case VI.B

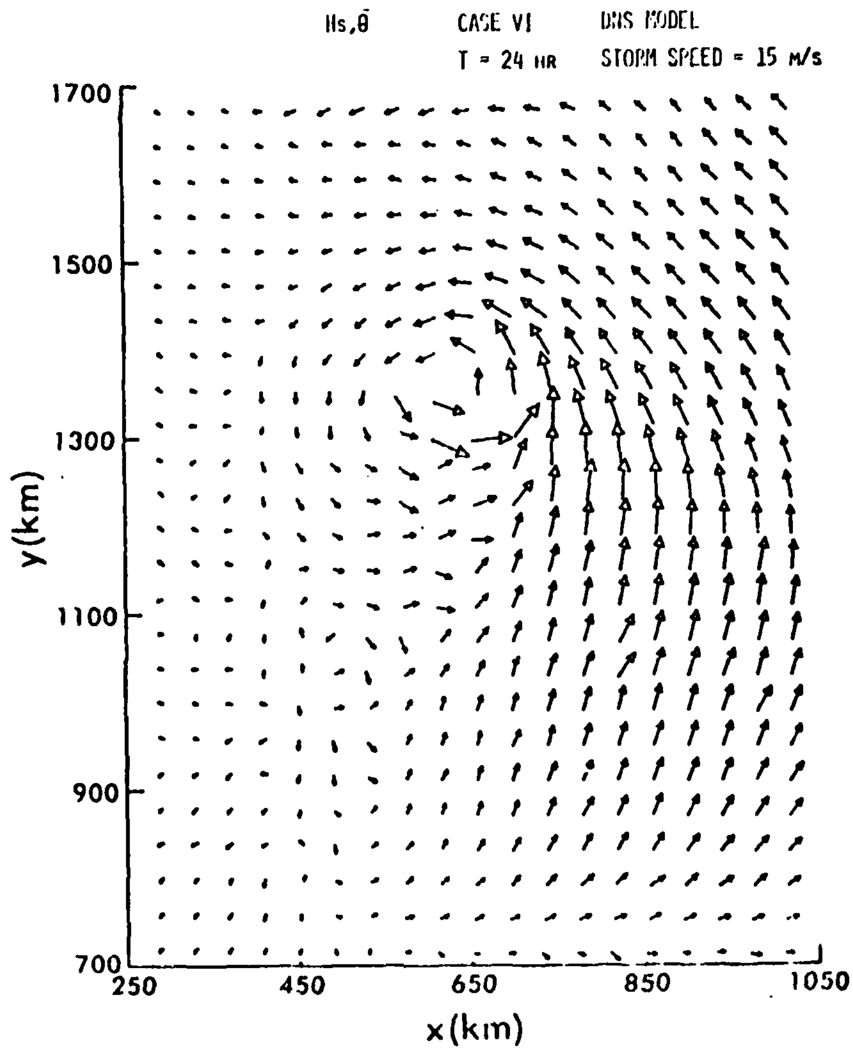


Figure 87. Model results, Case VI.B

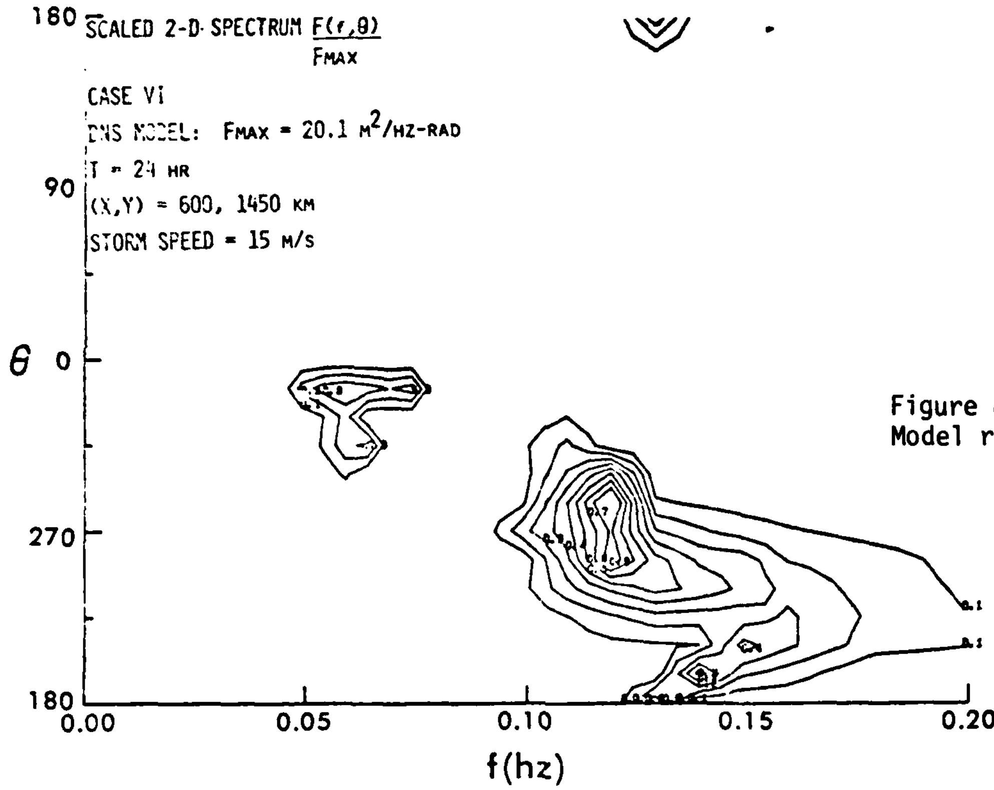


Figure 88.
Model results, Case VI.B

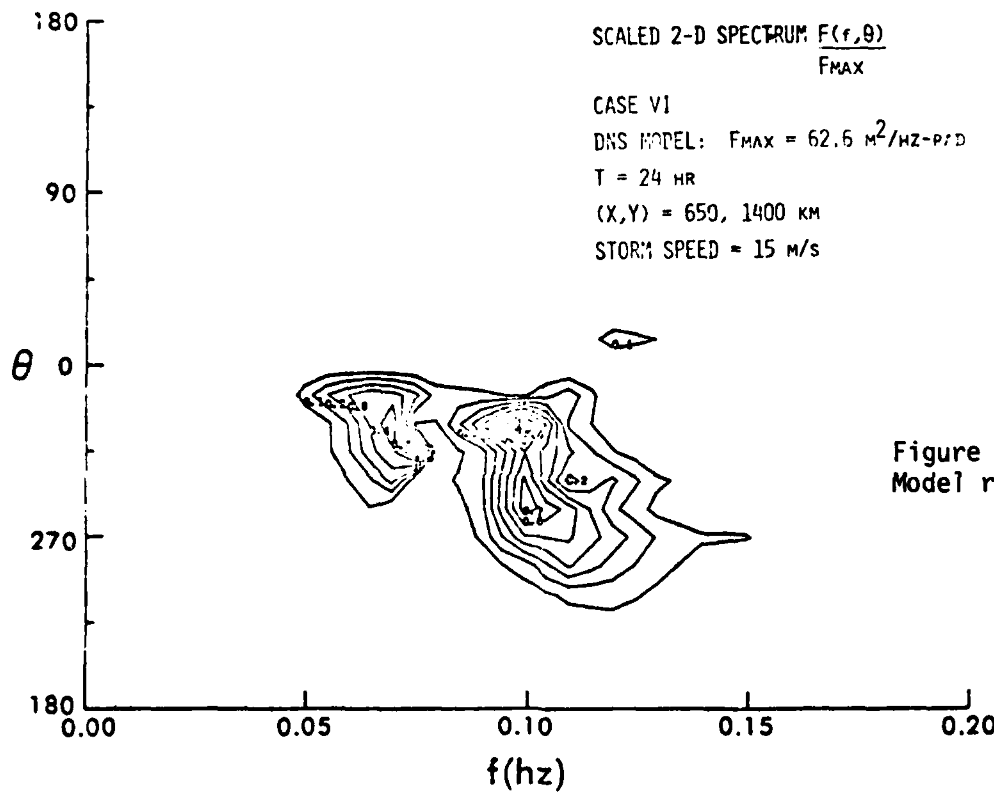


Figure 89.
Model results, Case VI.B

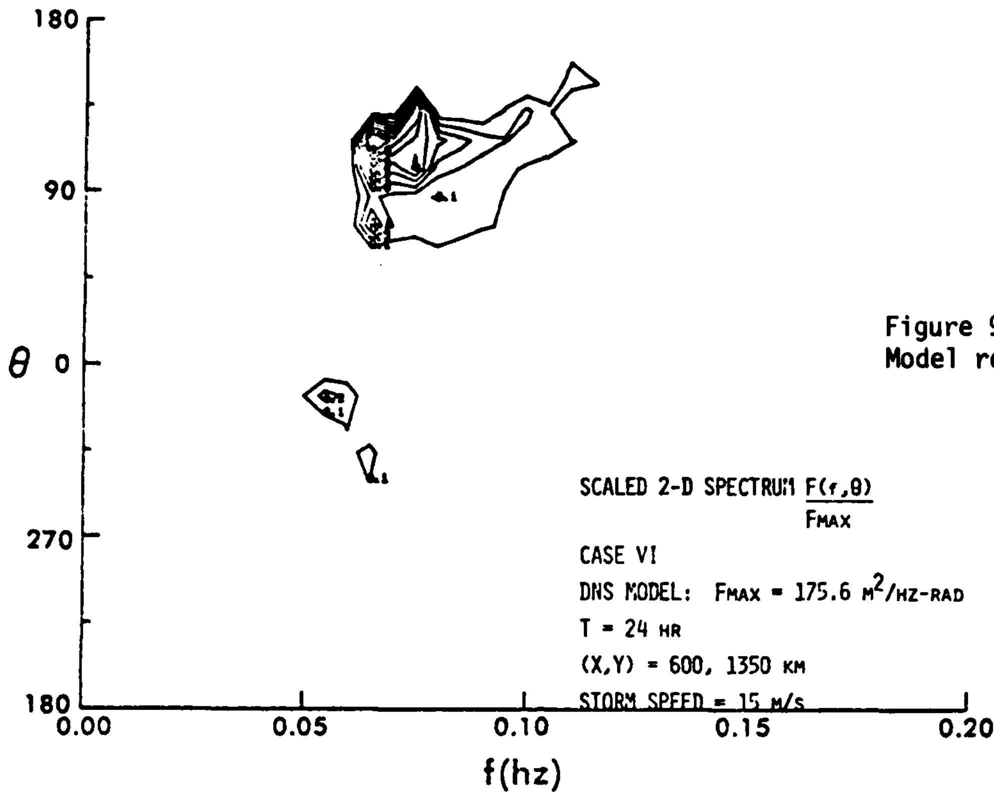


Figure 90.
Model results, Case VI.B

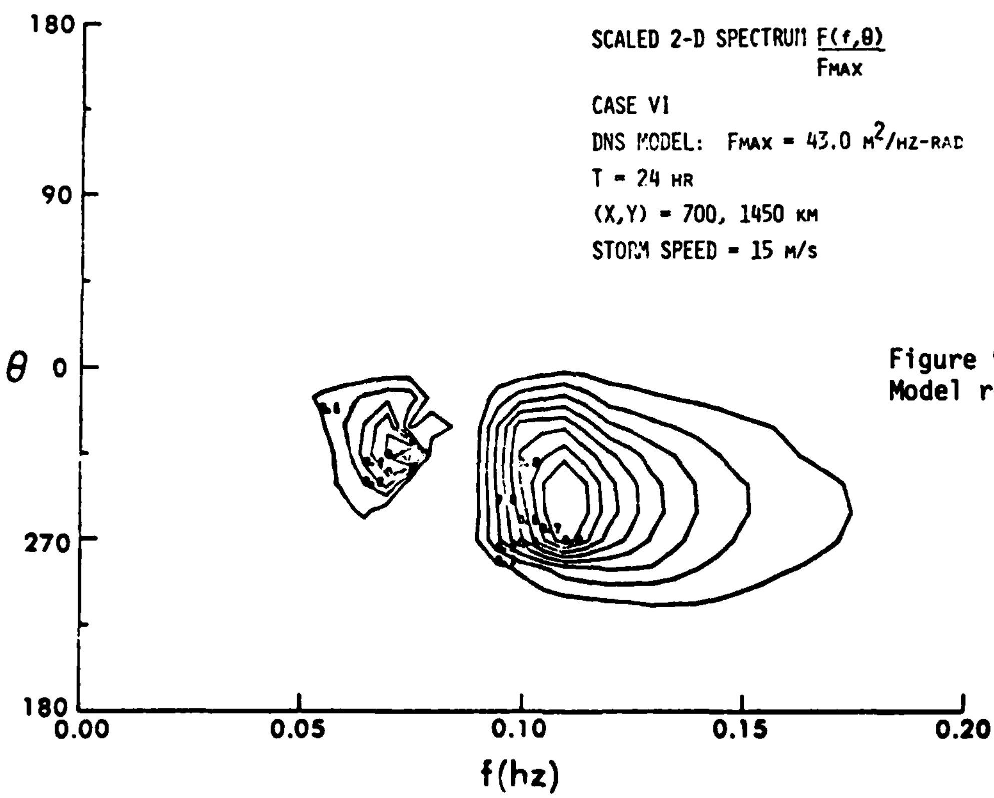


Figure 91.
Model results, Case VI.B

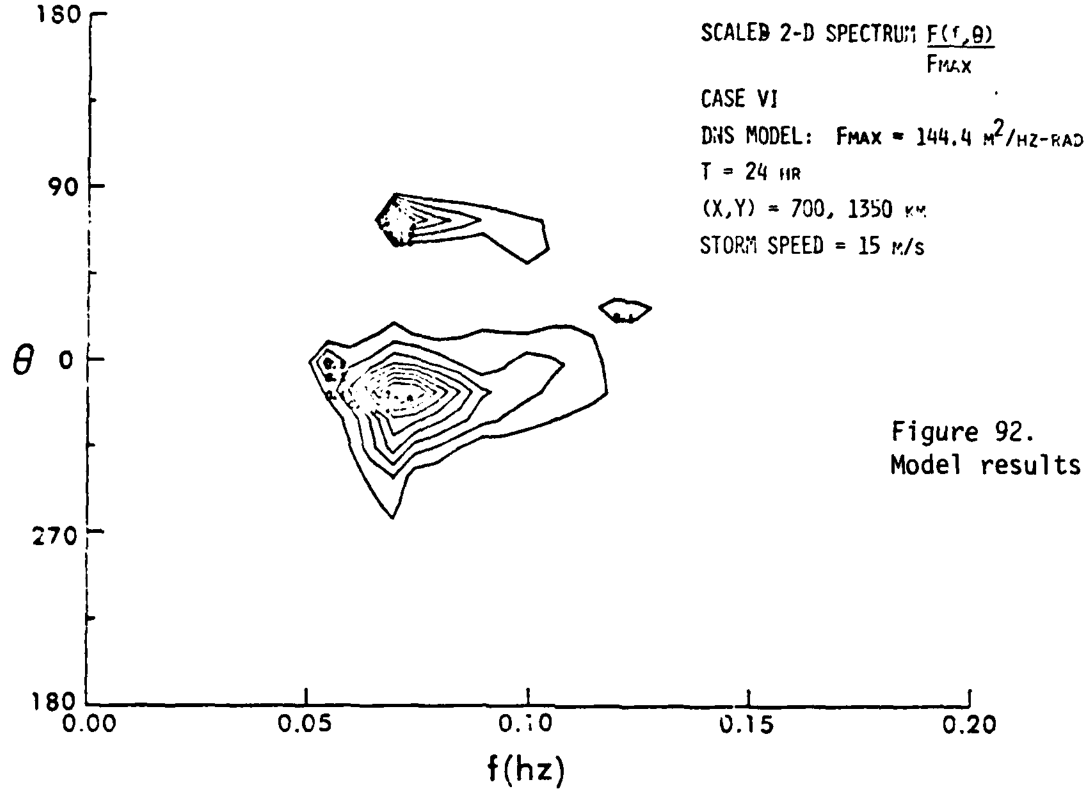


Figure 92.
Model results, Case VI.B

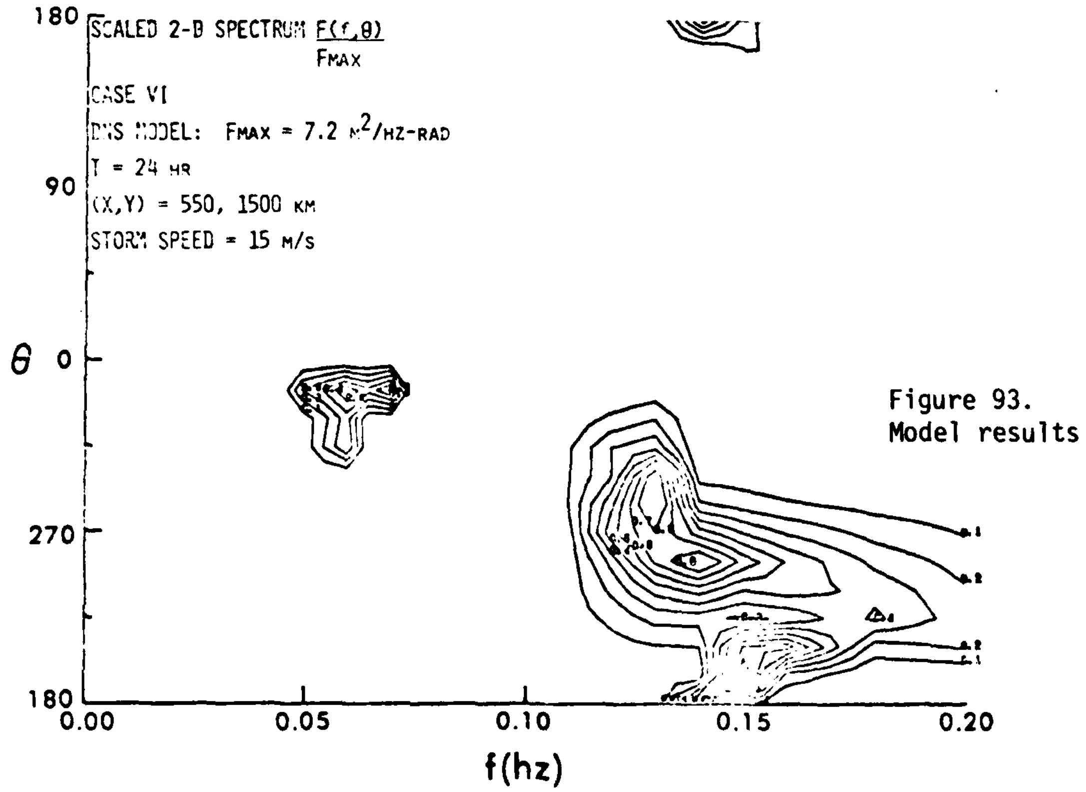


Figure 93.
Model results, Case VI.B

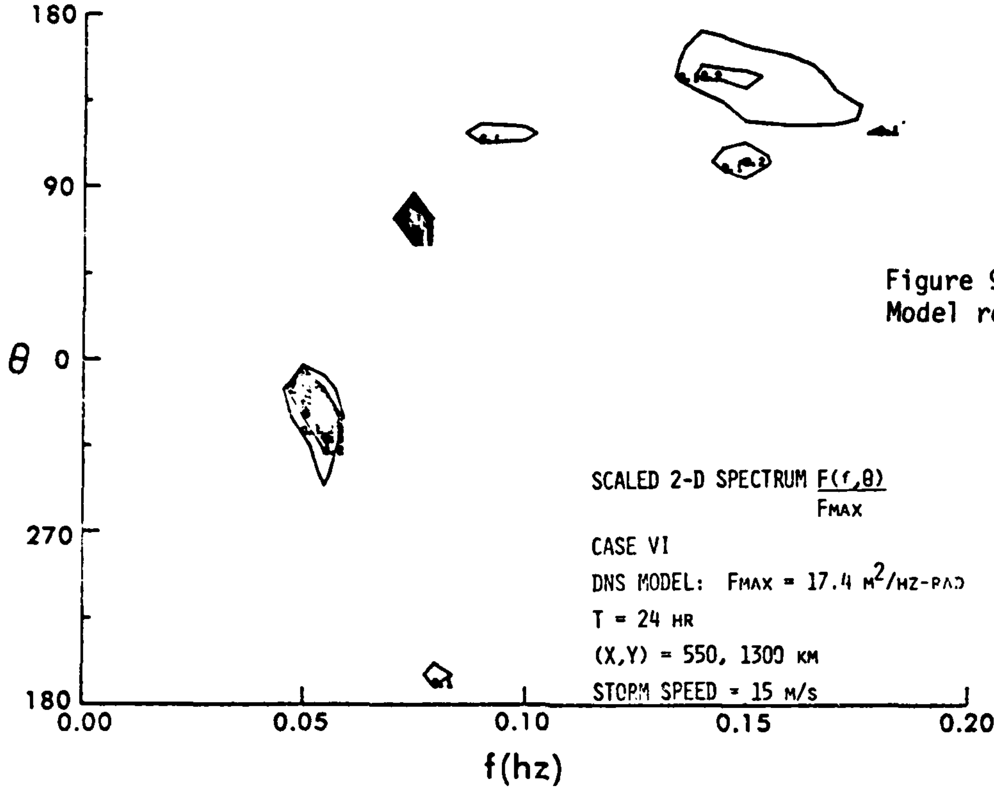


Figure 94.
Model results, Case VI.B

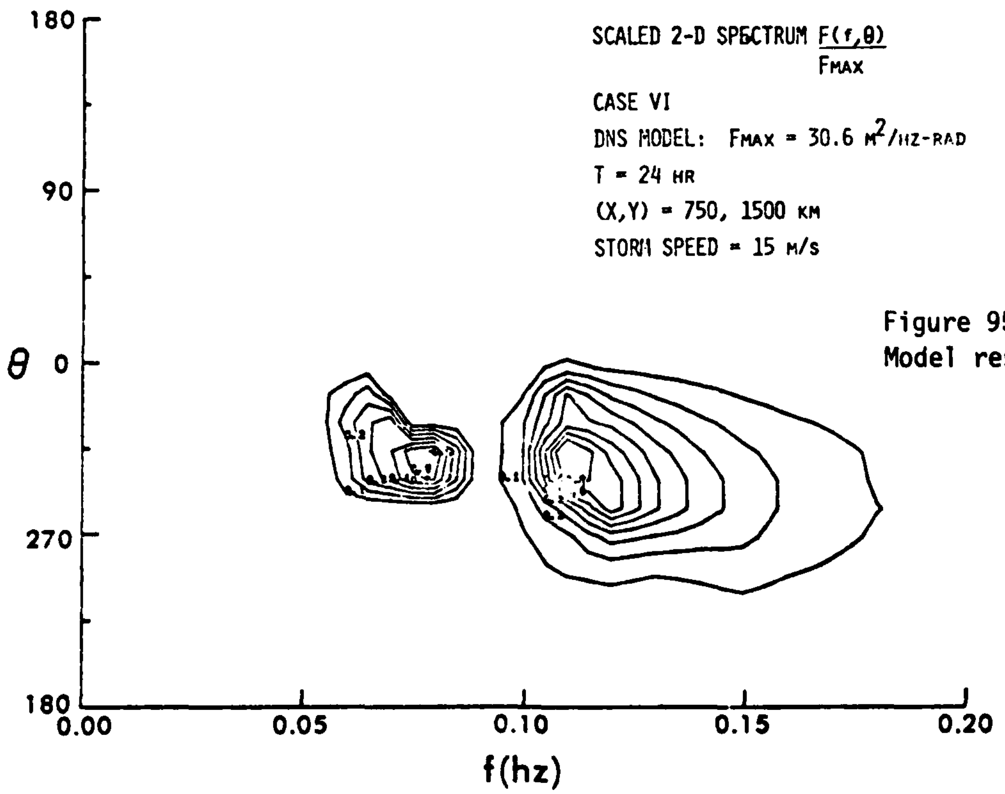


Figure 95.
Model results, Case VI.B

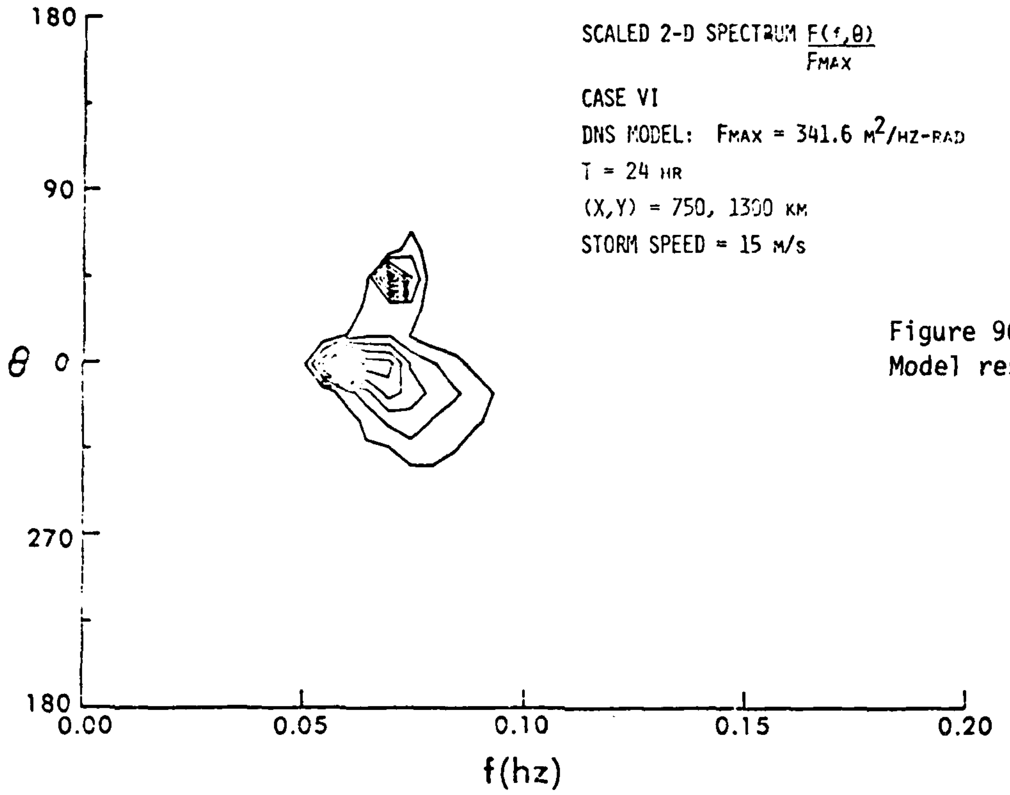


Figure 96.
Model results, Case VI.B

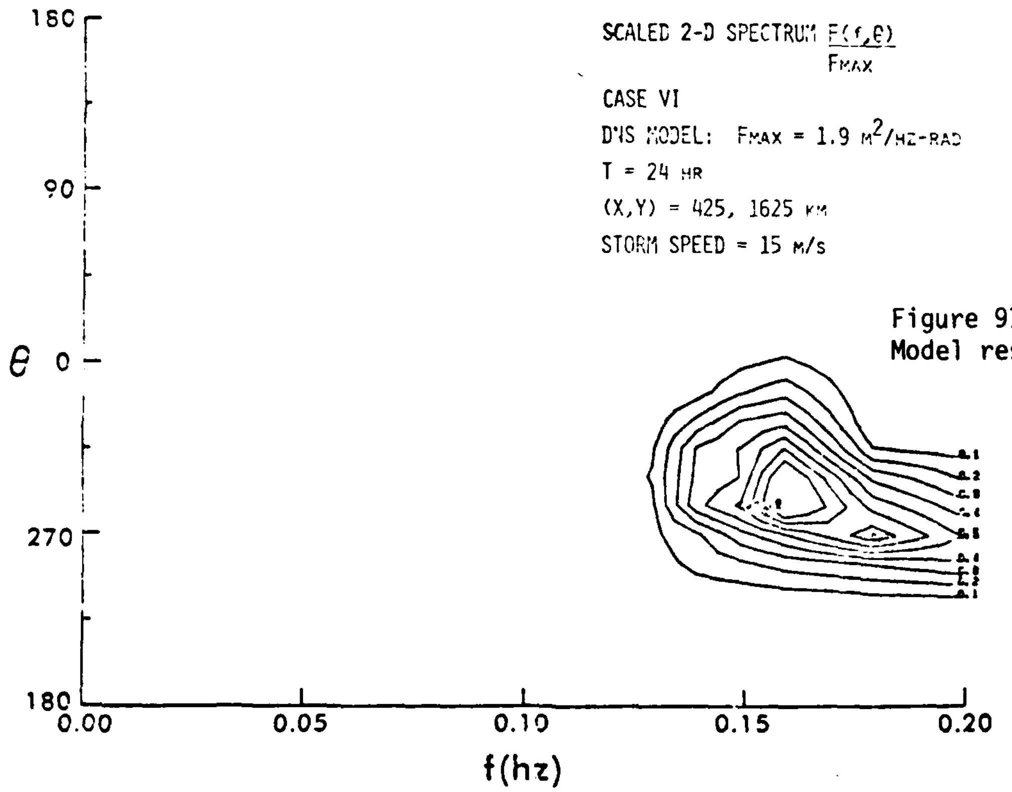


Figure 97.
Model results, Case VI.C

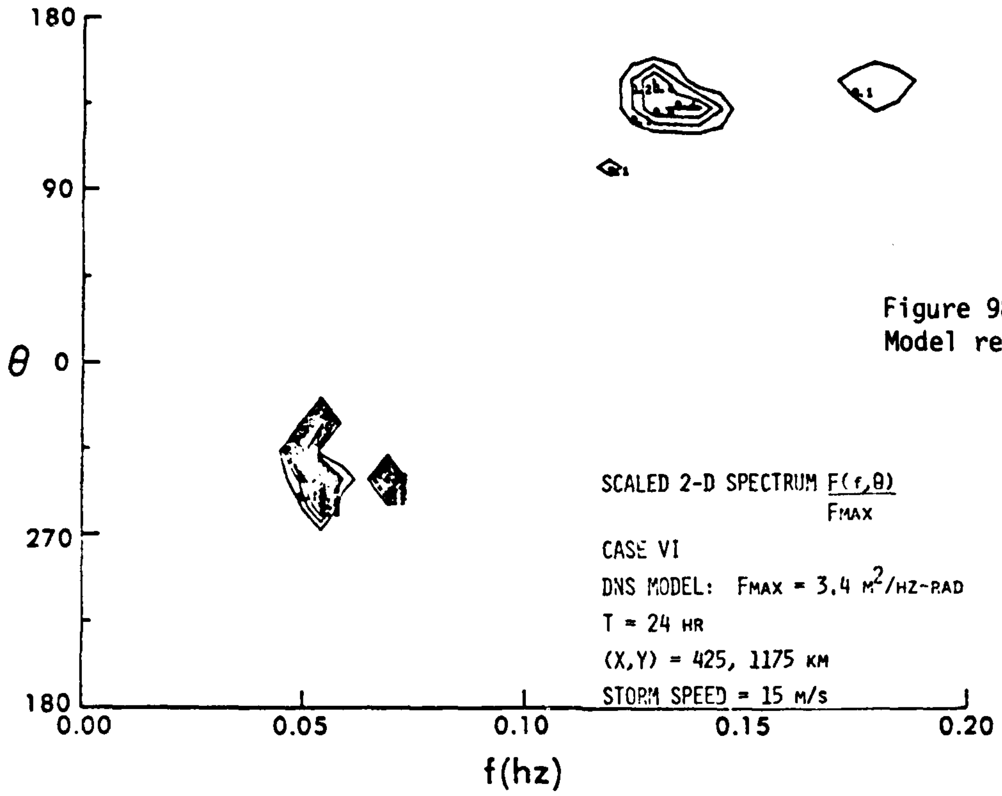


Figure 98.
Model results, Case VI.B

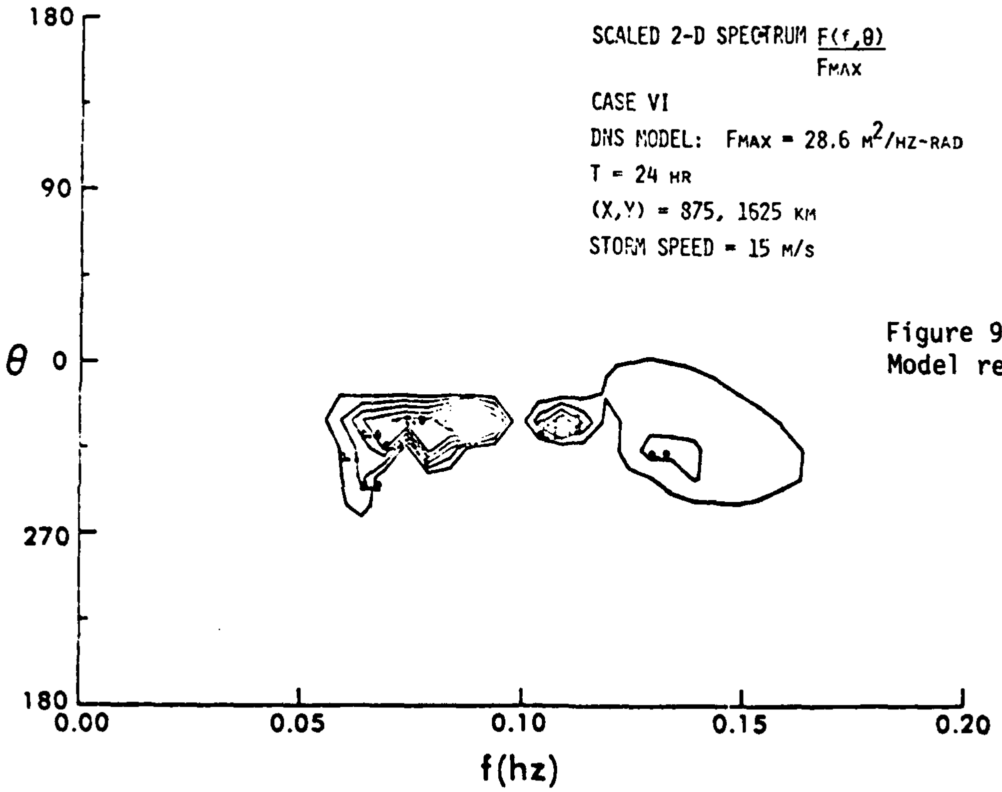


Figure 99.
Model results, Case VI.B

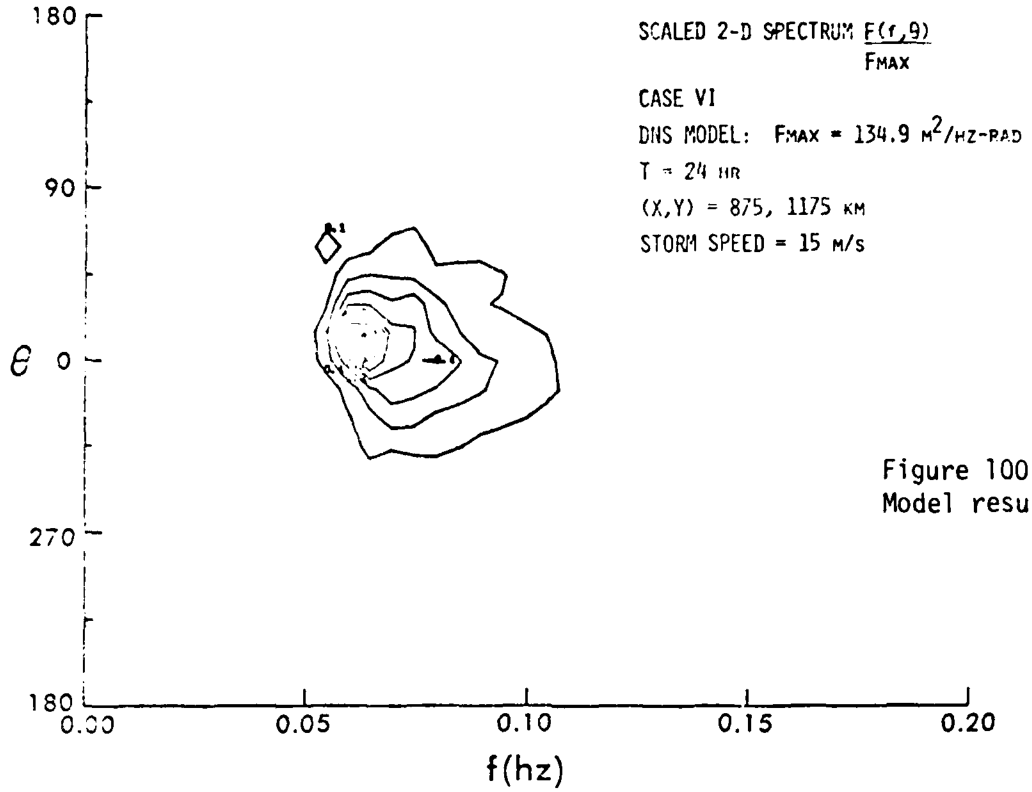


Figure 100.
Model results, Case VI.B

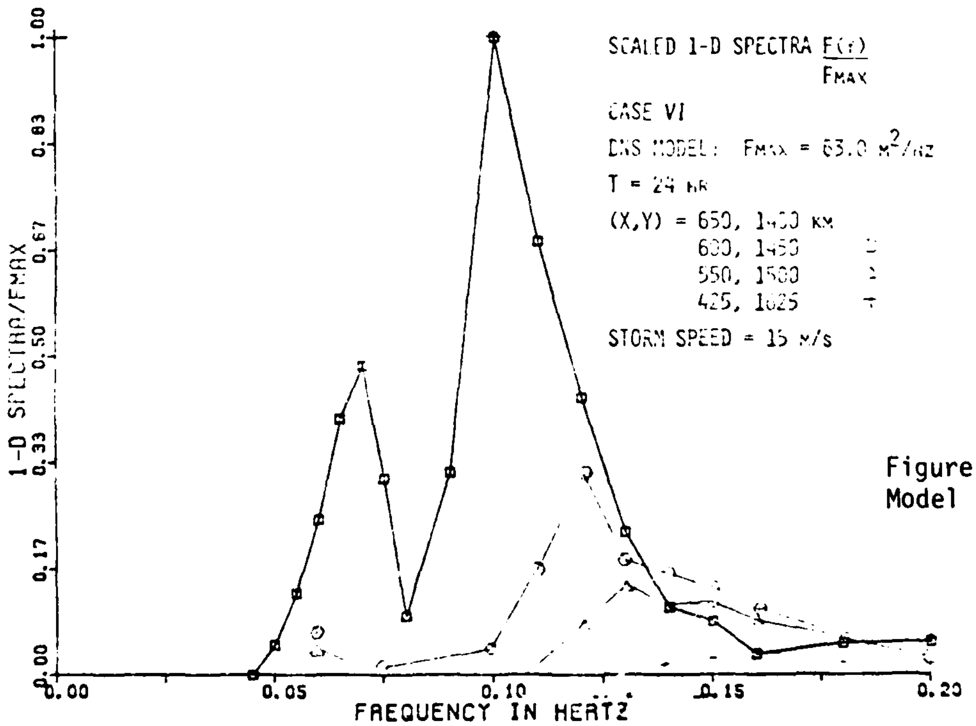


Figure 101.
Model results, Case VI.B

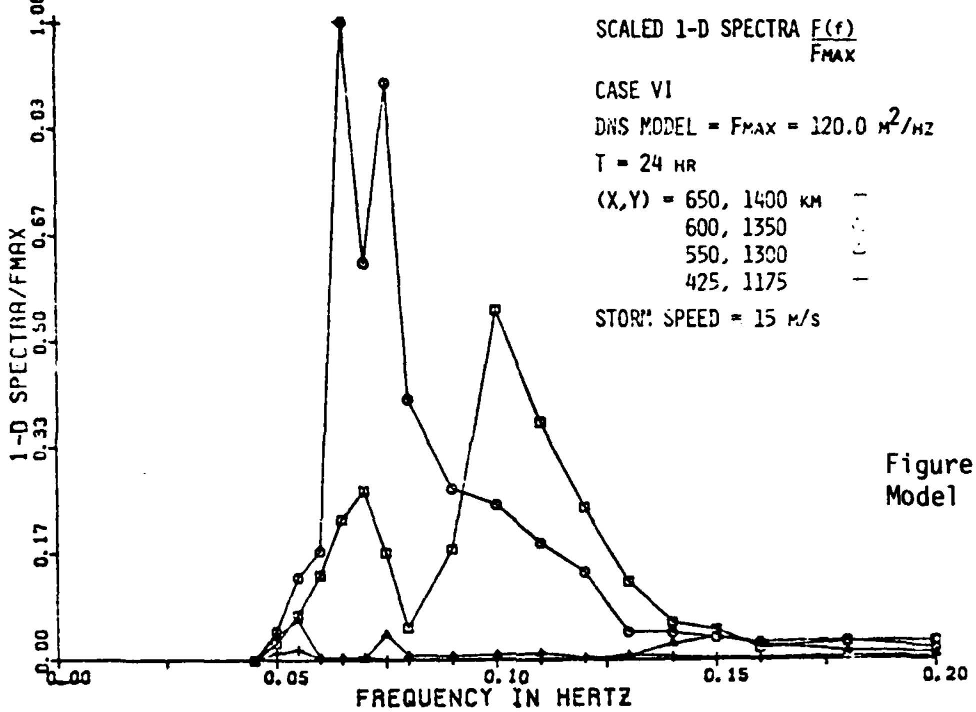


Figure 102.
Model results, Case VI.B

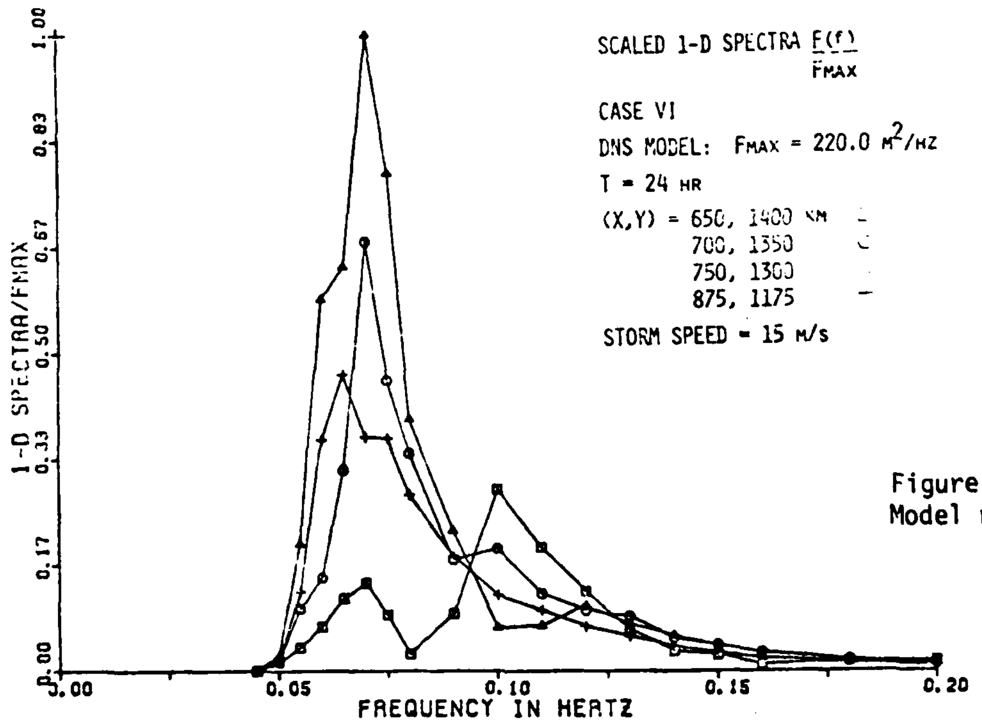


Figure 103.
Model results, Case VI.B

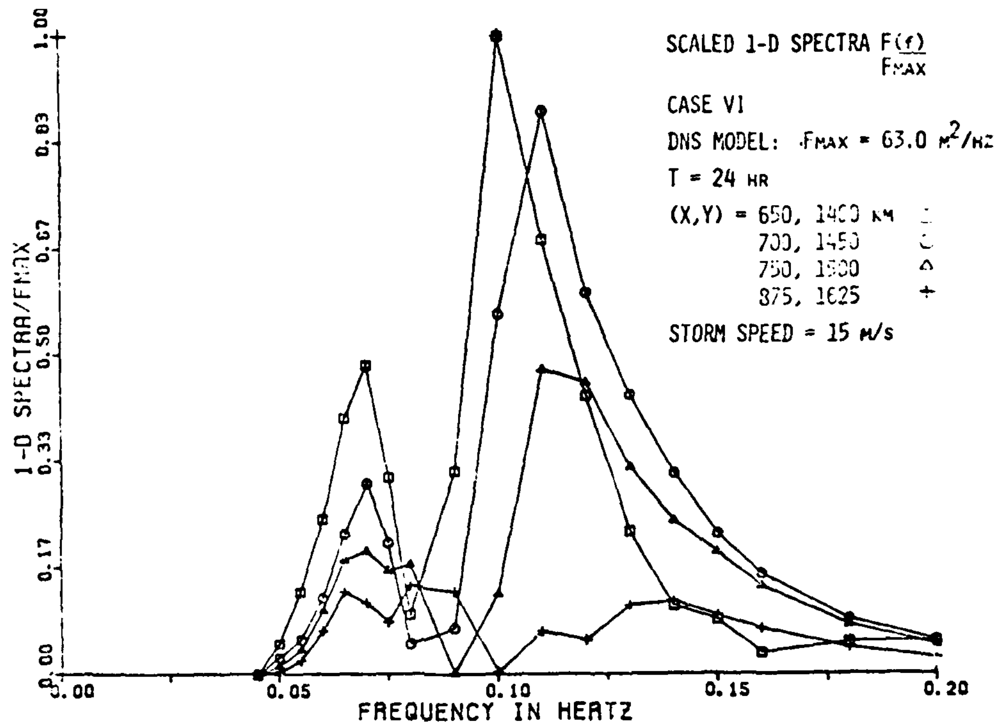


Figure 104. Model results, Case VI.B

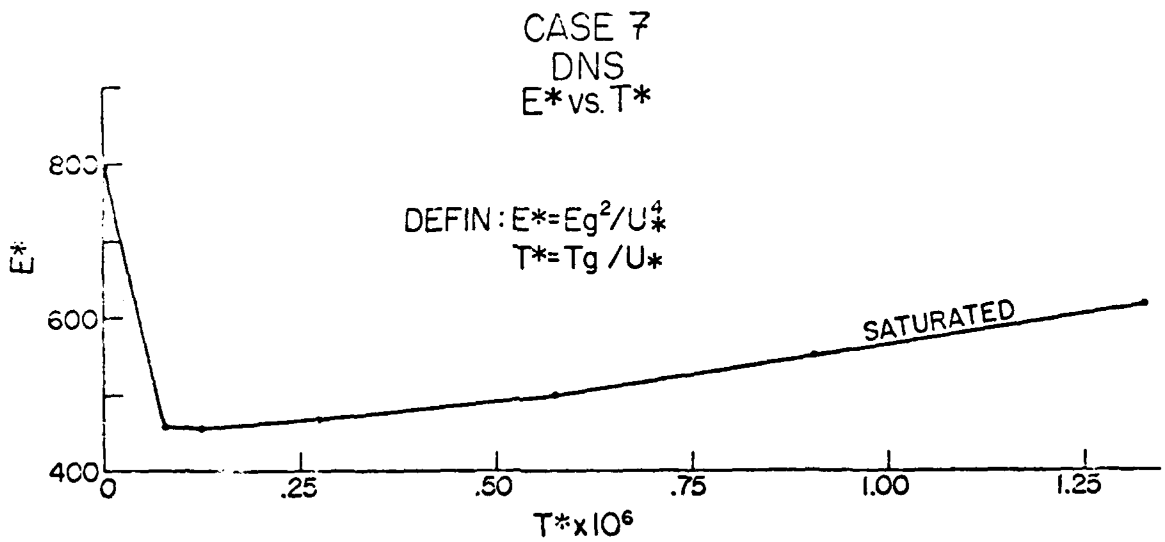


Figure 105. Model results, Case VII

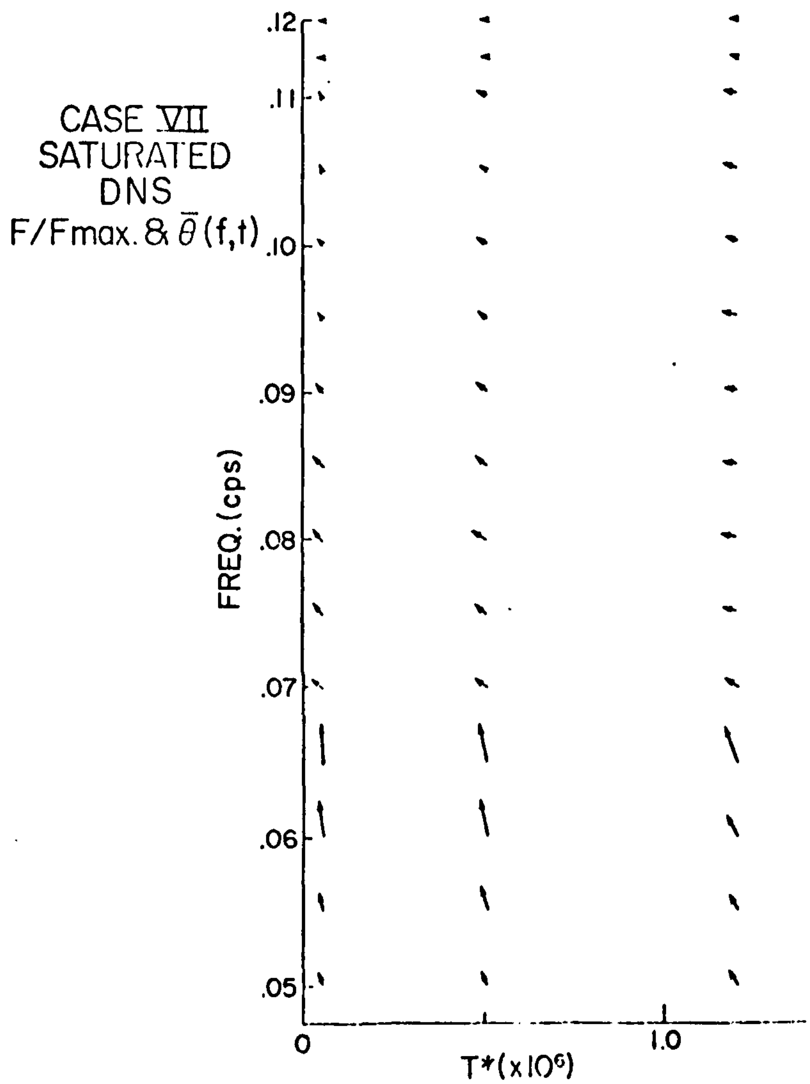


Figure 106. Model results, Case VII

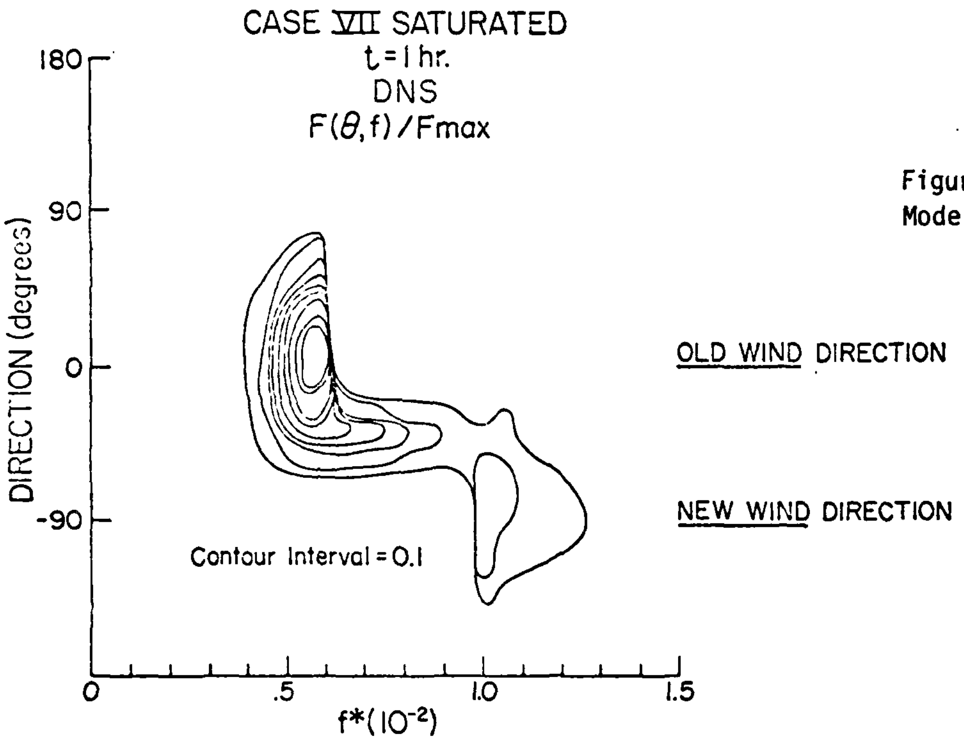


Figure 107.
Model results, Case VII

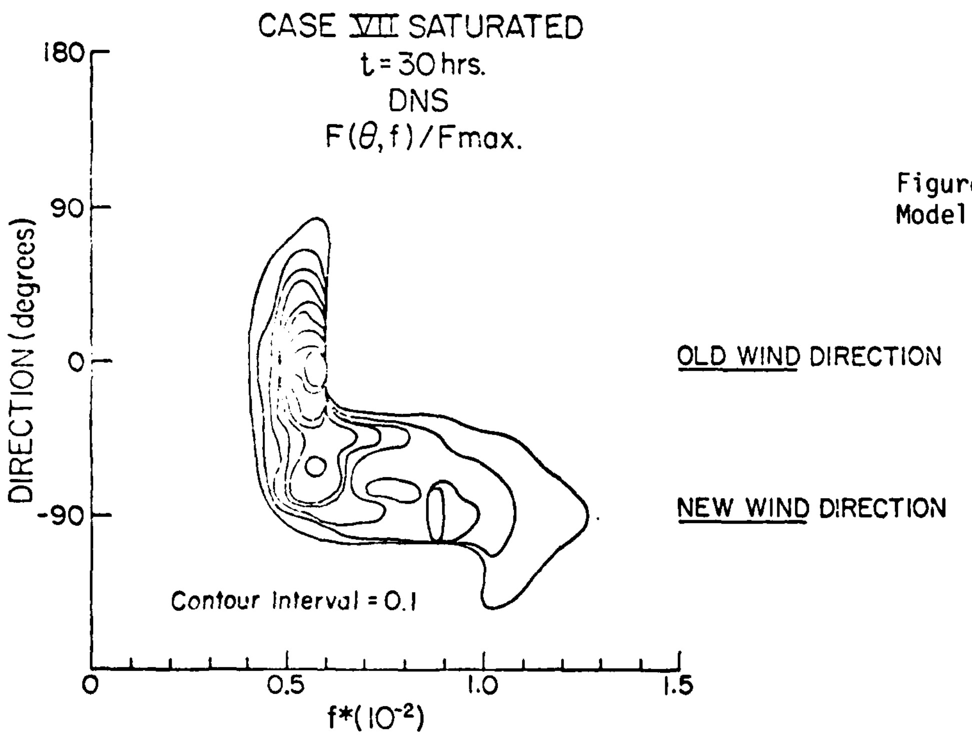


Figure 108.
Model results, Case VII

APPENDIX

A Summary of Calcomp-Compatible Plot Programs for the Wave Model

Isaac Traxler

Computer Sciences Corporation

NSTL Station, MS 39529

A. INTRODUCTION

The following documentation is divided up into sections. Each major section is a separate program. Each program includes a short text description, an input data description, a variable definition section, a program listing, and an example plot. The variable list is included with each one instead of overall because the meaning of variables change. Unfortunately the programs were not designed with a global variable usage plan in mind.

Whenever possible subroutines were borrowed from one program to make another program easier. Most of the tape I/O is done in READS and READJS. All the contour plots are created by passing parameters to CONTUN. CONTUN is a contourplot creating subroutine that has obviously been implemented on several different machines using several different plot packages. The included version works on Calcomp compatible systems.

The plot programs, like the wave model, run on a CDC 6600 using a 60-bit word. The programs are dependent on the word size because of the data tape. The data is packed on the tape two values per sixty bits. This fact causes the programs to be machine dependent since they must separate out the data. All text is used assuming 10 characters per word may be stored. Only changes to dimensions and subscripts would be necessary to alter this.

The following programs allow some of the data produced by the DNS WAVE Model to be graphically displayed. These programs evolved over a long period of time. At first, a program was written that produced a few simple curve plots. Their original purpose was to allow the results of the model to be interpreted easier and more fully. One by one separate sections were added to this and eventually contour plotting programs were written. As time passed these programs were altered continuously to extract more information about the model. At some point the Wave MODEL Intercomparison Workshop became a reality. Shortly thereafter, a preliminary version Wave Modeler's Newsletter was released. It contained a list of "standard" plots that each participant should attempt to produce. These programs began to drift in that direction. Whenever time permitted they were altered so that they would correspond to the plot requirements.

As time went by several newsletters were written. Some of them contained new plot requirements. "Corrections" were constantly being issued to the existing requirements. New programs were written to produce these plots. Old programs were modified. The final result is the following collection of programs. They were never created with the idea of a plot package in mind. They simply evolved to meet the changing requirements.

With a certain amount of study, these programs can be altered to produce an extremely large variety of plots. Before attempting to change them or even use them, I recommend that the user become very familiar with the format of the history and summary tapes.

B. PROGRAM PLOTWM

This program produces six different plots. The plots are chosen via a plot type number followed by appropriate data cards. Some of the plots have the ability to display multiple curves that are user selectable. All plots can be run separately or in conjunction with others. A plot may be run several times in one execution run. The following is a list of plots that the program can produce: Significant Wave Height, Peak Frequency, Mean Direction, 1-D Spectrum, and 2-D Spectrum.

The significant Wave Height plot displays a theorectical curve and an observed curve (program calculated) on an axis system of Significant Wave Height vs. fetch. The axes are labeled and provide for a title as well as providing for an overall title at the top of the graph.

The Peak Frequency section produces a graph with two curves, one theoretical and one observed. The X-axis corresponds to fetch and the Y-axis corresponds to frequency. Axis labeling, axis titling and overall titling is also provided.

The Mean Direction section plots a single curve, mean direction, with the axes being direction and fetch. The same labeling capabilities exist for this routine as above.

1-D Spectrum plots can display multiple curves on one graph. Three different routines exist to do 1-D Spectrum plots. The first one does 1-D Spectrum plots at final time step with grid point as family parameter. A second section allows time as a family parameter and uses only one grid point. A third section of code allows any single time to be chosen and uses grid point as a parameter. This one was written so that plots could be made that were not at the final time step. It is slower than the first since it uses the history tape rather than the summary tape.

The 2-D Spectrum section produces a graph similar to the others in format using direction as the family parameter. Any time step is allowable.

This program has two primary sections. The first section requests primary data and takes care of most overhead. The second part of the program requests a plot type number and then turns control over to each separate section of the program. Each plot type requires further input. At the end of the section the program returns to requesting a plot type number.

INPUT DATA LIST FOR PLOTWM

FORMAT	VARIABLE	DESCRIPTION
7I10	FACMAIN,FAC1,FAC2, FAC3,FAC4,FAC5, FAC6	FACMAIN is the overall scale factor for all plots, the other values are multiplied times FACMAIN for each particular plot, FAC1 corresponds with plot type 1, FAC2 corresponds with plot type 2, etc., FAC5 is also used for plot type 7.
I10	NGRID	Number of grid points wave model used in generating data tape.
8I10	NF,(FREQ(j), j=1,NF,1)	NF is number of frequencies, FREQ is an array containing a list of frequencies.
I10,7F10.0 /,(8f10.0)	ND,(DIR(k), k=1,ND,1)	ND is number of directions, DIR is direction list.
2F10.0	U,D	U is wind speed and D is wind direction
I10,10x, 3(i10,F10.0), (4(I10,F10.0)	NS,(NOSTA(i), Y(i), i=1,NS,1)	NS is number of stations, NOSTA is station list, Y is a list of fetches, each Y(i) is Associated with NOSTA(i).
2F10.0	SF(1),SF(2)	Scale factors
2I10	ILP,IPL	If ILP&IPL both equal 0 then default values used, otherwise 0 implies false and 1 implies true.
I10	IPTYPE	If IPTYPE=0 then program stops, values 1 thru 7 select plot types.
PLOT TYPE 1	SIGNIFICANT WAVE HEIGHT	
8A10	TITLE(i),i=3,10	Title of plot, appear across top of plot
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is number of characters in X-axis label, XLABEL is the label of X-axis.
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is number of characters in Y-axis label, YLABEL is the Y-axis label.
PLOT TYPE 2	PEAK FREQUENCY PLOT	
8A10	TITLE(i),i=3,10	TITLE is label displayed at top of plot.
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is number of characters in XLABEL, XLABEL is the X-axis label.
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is the number of characters in Y-axis label, YLABEL is the Y-axis label.

INPUT DATA LIST FOR PLOTWM (Cont'd)

PLOT TYPE 3 MEAN DIRECTION PLOT

FORMAT	VARIABLE	DESCRIPTION
8A10	TITLE(i),i=3,10	TITLE is the title of the plot.
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is the number of characters in XLABEL, XLABEL contains actual X-axis label.
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is the number of characters in the Y-axis label, YLABEL is the Y-axis label.

PLOT TYPE 4 1-D SPECTRUM

8A10	TITLE(i),i=3,10	TITLE is title of plot.
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is the number of characters in XLABEL, XLABEL is the X-axis label
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is the number of characters in YLABEL, YLABEL is the Y-axis label
I10	NDU	If NDU=1 then non-dimensional units are used, else dimensional units are used.
I10	NSID	Number of stations to be plotted.
8I10	NOID(i),i=1,NSID,1	Station list, up to ten station numbers to be plotted.

PLOT TYPE 5 1-D SPECTRUM

8A10	TITLE(i),i=3,10	TITLE is the title of the plot
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is the number of characters in XLABEL, XLABEL is the X-axis label
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is the number of characters in YLABEL, YLABEL is the Y-axis label
I10	NDU	If NDU=1 then non-dimensional units are used, else dimensional units are used
8I10	NSTEP,(ITIME(i), i=1,NSTEP,1)	NSTEP is number of time steps to be plotted, ITIME is a list of time steps to be plotted.
I10	INGRDPT	The grid point number to be plotted.

INPUT DATA LIST FOR PLOTWM (Cont'd)

PLOT TYPE 6 — 2-D SPECTRUM

FORMAT	VARIABLE	DESCRIPTION
8A10	TITLE(i),i=3,10	TITLE is the title of the plot.
I10,10x,6A10	NXCHAR,XLABEL	NXCHAR is the number of characters in XLABEL, XLABEL is the X-axis label.
I10,10x,6A10	NYCHAR,YLABEL	NYCHAR is the number of characters in YLABLE, YLABEL is the label of the Y-axis.
I10	INGRDPT	The grid point number that is to be plotted.
I10	ITSTEP	The time step number to be plotted.
8I10	NSD,(IDIR(i),i=1, NSD,1)	NSD is the number of directions to be plotted, IDIR is the array of directions

PLOT TYPE 7 — 1-D SPECTRUM

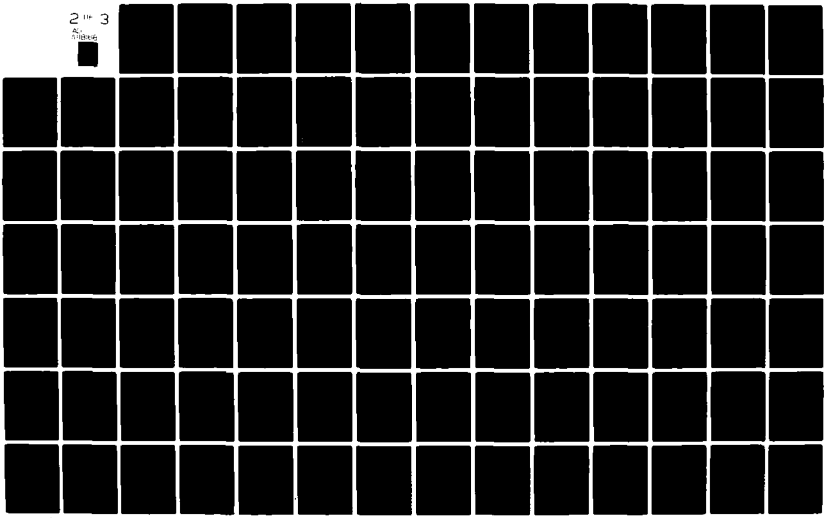
I10	NSTEP	Time step to be plotted.
I10	NGRDPTS	Number of grid points to be plotted.
8I10	GRDPTS(i),i=1, NGRDPTS	Array containing list of grid points to be plotted.
I10,7A10	NCHAR,TITLE(i) i=3,9	NCHAR is number of characters in TITLE,TITLE is the title of the plot.
I10,7A10	NXCHAR,XLABEL	NXCHAR is the number of characters in XLABEL, XLABEL is the X-axis label.
I10,7A10	NYCHAR,YLABEL	NYCHAR is the number of characters in YLABEL, YLABEL is the label of the Y-axis.
I10	NDU	If NDU=1 then non-demensional units are used else dimensional units are used.
I10	NSTEP	Time step selected.

VARIABLE LIST FOR PLOTWM

SS(2)	-	Real, scale factors, hold default values of 1.0
FREQ(20)	-	Real, array containing list of frequencies in hz
DIR(36)	-	Real, array holding directions in degrees
Y(100)	-	Real, array containing a list of fetch values to correspond to station list
SF(2)	-	Real, actual scale factors, used to scale Y(fetch)
BUFFER(1511)	-	Real, used as input buffer to hold SREC off of summary tape
CURVE(100,10)	-	Real, stores dependent variable values to be plotted, allows up to 10 curves with a maximum of 100 points each to be plotted
TITLE(10)	-	Real (used as character), contains title of plot
XLABEL(6)	-	Real (used as character, contains label of X-axis
YLABLE(6)	-	Real (used as character), contains Y-axis label
S1D(500,20)	-	Real, input buffer to hold 1-D spectrum off JSREC on summary tape
FM(500)	-	Real, contains Peak Frequency for each station
TAR(100)	-	Real, temporary array used to hold independent variable values for plotting
NSTAR	-	Real, corresponds to N* in Wave Modeler's newsletter, normalization value
USTAR	-	Real, 100*NSTAR
BFPM	-	Real, corresponds to $F_{PM}(f_{PM})$ in newsletter
G	-	Real, gravity
NOSTA(100)	-	Integer, list of station numbers to correspond to Y(list of fetches)
LUN(2)	-	Integer array to hold logical unit numbers
NO1D(10)	-	Integer, used to hold list of station numbers to be plotted
NB	-	Integer, size of BUFFER, number of values to read in
NX	-	Integer size of line printer plot in X direction
NY	-	Integer, size of line printer plot in Y direction

AD-A118 168 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 20/4
APPLICATION OF A DISCRETE NONLINEAR SPECTRAL MODEL TO IDEAL CAS--ETC(U)
APR 82 J H ALLENDER, M LYBANON
UNCLASSIFIED NORDA-TN-148 NL

2 OF 3
AD
A118168



VARIABLE LIST FOR PLOTWM (Cont'd)

NF	-	Integer, number of frequencies
ND	-	Integer, number of directions
NGRID	-	Integer, number of grid points
NEWPAGE	-	Logical, flag indicating whether or not to do a new page of plots
SREC	-	Logical, true if SREC has been loaded into BUFFER
JSREC	-	True if JSREC has been loaded into SID
LP	-	Logical, true means do a printer plot, false means do not do a printer plot
PL	-	Logical, flag for plotter plot
FACMAIN	-	Real overall scaling factor for plots, used as a multiplier in every call to FACTOR
FAC1	-	Real, scale factor for plot type 1, multiplied by FACMAIN for overall factor
FAC2	-	Real, scale factor for plot type 2, multiplied by FACMAIN for overall factor
FAC3	-	Real, scale factor for plot type 3, multiplied by FACMAIN for overall factor
FAC4	-	Real, scale factor for plot type 4, multiplied by FACMAIN for overall factor
FAC5	-	Real, scale factor for plot type 5, multiplied by FACMAIN for overall factor
FAC6	-	Real, scale factor for plot type 6, multiplied by FACMAIN for overall factor
U	-	Real, wind speed
D	-	Real, direction
C1, C2, C3	-	Real, used to hold temp results of expressions
ILP	-	Integer, used to indicate how LP should be set
IPL	-	Integer, used to indicate how PL should be set; if ILP and IPL are 0 then default values are used for LP and PL, otherwise a 1 means true and 0 means false
IPTYPE	-	Integer, plot type

<u>VARIABLE LIST FOR PLOTWM</u> (Cont'd)

IOFSET	-	Integer, used to indicate how many words to skip on input of record
NDU	-	Integer, flag for non-dimensional units

PROGRAM PLOTHM

```
      PROGRAM PLOTHM(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
     +    TAPE1, TAPE2,    TAPE7)
C
C     PROGRAM TO GENERATE PLOTS OF BARNETT WAVE MODEL RESULTS.
C
      REAL SS(2), FREQ(20), DIR(36), Y(100), SF(2), BUFFER(1511)
      REAL CURVE(100,10), TITLE(10), XLABEL(6), YLABEL(6)
      REAL S10(500,20), FM(500)
      REAL TAR(100)
C TAR - TEMPORARY ARRAY USED TO HOLD X-AXIS VALUES IF THEY HAVE BEEN AL
      REAL NSTAR
C NSTAR - CORRESPONDS TO N* USED IN CALCULATIONS
      INTEGER NOSTA(100), LUN(2), NOID(10)
      LOGICAL NEWPAGE, SREC, LP, PL
      LOGICAL USREC
      DATA G, SS, NB /9.81, 2*1., 1511/
      DATA NCHAR, TITLE(1), TITLE(2) /100, 2*10H          /
      DATA NX, NY /120, 60/
      DATA LUN /1,2/
      DATA NEWPAGE, SREC, LP, PL /.TRUE., .FALSE., .FALSE., .TRUE./
      DATA USREC /.FALSE./
      DATA NSTAR/0.855/
      DATA USTAR/85.5/,BFPM/1351000.0/
C
C  FACMAIN - MAIN SCALING FACTOR FOR PLOTS
C  FAC1 - SCALING FACTOR FOR PLOT TYPE 1
C  FAC2 - SCALING FACTOR FOR PLOT TYPE 2
C  FAC3 - SCALING FACTOR FOR PLOT TYPE 3
C  FAC4 - SCALING FACTOR FOR PLOT TYPE 4
C  FAC5 - SCALING FACTOR FOR PLOT TYPE 5
      CALL PLOTS(0,0,7)
 1006 FORMAT(8F10.0)
      READ(5,1006) FACMAIN,FAC1,FAC2,FAC3,FAC4,FAC5,FAC6
C     READ MODEL CONDITIONS.
      READ (5,1000) NGRID
      READ (5,1000) NF, (FREQ(J),J=1,NF)
      READ (5,1000) ND, (DIR(K),K=1,ND)
 1000 FORMAT (I10,7F10.0/(8F10.0))
      READ (5,1002) U, D
      WRITE (6,2000) NGRID
 2000 FORMAT (1H1,45X31HBARNETT WAVE MODEL PLOT PROGRAM//
     +        52X16HMODEL CONDITIONS//
     +        I14,12H GRID POINTS)
      WRITE (6,2001)
 2001 FORMAT (/11X11HFREQUENCIES)
      WRITE (6,2002) (FREQ(J),J=1,NF)
 2002 FORMAT (1X12F10.3)
      WRITE (6,2003)
 2003 FORMAT (/11X10HDIRECTIONS)
      WRITE (6,2002) (DIR(K),K=1,ND)
      WRITE (6,2004)
 2004 FORMAT (/11X24HWIND SPEED AND DIRECTION)
      WRITE (6,2002) U, D
      READ (5,1001) NS, (NOSTA(I), Y(I), I=1,NS)
 1001 FORMAT (I10,10X3(I10,F10.0)/(4(I10,F10.0)))
      READ (5,1002) SF
 1002 FORMAT (2F10.0)
      IF (SF(1) .LE. 0.) SF(1) = SS(1)
      IF (SF(2) .LE. 0.) SF(2) = SS(2)
      WRITE (6,2005)
 2005 FORMAT (//54X15HPLOT PARAMETERS)
      WRITE (6,2006)
 2006 FORMAT (/11X7HFETCHES)
      WRITE (6,2002) (Y(I),I=1,NS)
      WRITE (6,2007)
 2007 FORMAT (/11X35HDISTANCE AND VELOCITY SCALE FACTORS)
      WRITE (6,2002) SF
      U = U * SF(2)
      WRITE (6,2008)
 2008 FORMAT (//22X19HAFTER SCALING . . .)
      WRITE (6,2006)
      WRITE (6,2002) (Y(I),I=1,NS)
      WRITE (6,2004)
      WRITE (6,2002) U, D
      WRITE(6,2009)FACMAIN,FAC1 ,FAC2,FAC3,FAC4,FAC5,FAC6
```

PROGRAM PLOTHM

```
 2009 FORMAT(8H FACMAIN,3X,4HFAC1,5X,4HFAC2,5X,4HFAC3,5X,4HFAC4,5X,
     +   4HFAC5,5X,4HFAC6,/,F7.3,1X,F8.4,5(2X,F7.3))
C SECTION FOR DIMENSIONED VARAIBLES
      C3 = U**2 / G
      C1 = 1.5E-3 * SQRT (C3)
      C2 = 2.84 * (G / U) * C3**0.3
C     APPLY SCALE FACTORS.
      DO 10 I=1,NS
      Y(I)=Y(I)*SF(1)
 10   CONTINUE
C     ILP = 1 CHOOSES LINEPRINTER PLOTS.
C     IPL = 1 CHOOSES ZETA PLOTS.
C     ILP = IPL = 0 CHOOSES DEFAULTS.
      READ (5,1005) ILP, IPL
 1005 FORMAT(8I10)
      IF (ILP .EQ. 0  .AND.  IPL .EQ. 0) GO TO 20
      IF (ILP .EQ. 0) LP = .FALSE.
      IF (ILP .EQ. 1) LP = .TRUE.
      IF (IPL .EQ. 0) PL = .FALSE.
      IF (IPL .EQ. 1) PL = .TRUE.
C     READ PLOT TYPE.
 20   READ (5,1000) IPTYPE
C
      IF(IPTYPE .LE. 0) CALL PLOT(0.0,0.0,999)
      IF (IPTYPE .LE. 0) STOP
C
      GOTO(30,130,230,330,430,530,630), IPTYPE
C     SIGNIFICANT WAVE HEIGHT PLOT.
 30   READ (5,1003) (TITLE(I),I=3,10)
 1003 FORMAT (8A10)
      READ (5,1004) NXCHAR, XLABEL
 1004 FORMAT (I10,10X6A10)
      READ (5,1004) NYCHAR, YLABEL
      IF (SREC) GO TO 40
      CALL READS (LUN(1), BUFFER, NB, NF)
C
      SREC = .TRUE.
 40   IOFSET = 11
C     CALCULATE OBSERVED AND THEORETICAL SIGNIFICANT WAVE HEIGHTS.
      DO 50 I=1,NS
      J = NOSTA(I) + IOFSET
      CURVE(I,1)=0.04*SQRT(BUFFER(J))
      CURVE(I,2) = C1 * SQRT (1000.0 * Y(I))
 50   CONTINUE
C     PLOT CURVES.
      IF (LP) CALL PLCURV(NS,2,Y,CURVE,0.0,0.0,0.0,0.0,
     +   NCHAR,TITLE, NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
      CALL FACTOR(FAC1*FACMAIN)
      IF(PL)CALL ZPLOT(NS,2,Y,CURVE,0.0,0.0,0.0,0.0,
     +    0.0,0.0,NCHAR,TITLE(3),
     +   NXCHAR,XLABEL,NYCHAR,YLABEL,10.1,6.5,0.0,10.1,0.0,6.5)
C
      GO TO 20
C     PEAK FREQUENCY PLOT.
  130 CONTINUE
C

      READ(5,1003) (TITLE(I),I=3,10)
      READ (5,1004) NXCHAR, XLABEL
      READ (5,1004) NYCHAR, YLABEL
      IF (JSREC) GO TO 140
C     READ JSREC.
      CALL READJS(LUN(1),S1D,NF,NGRID)
C
      JSREC = .TRUE.
 140  CONTINUE
C     CALCULATE OBSERVED AND THEORETICAL PEAK FREQUENCIES.
      CALL PKFREQ(FREQ,S1D,NS,NF,FM,NOSTA)
C
      DO 150 I=1,NS
C  FM(I) IS THE PEAK FREQUENCY FOR STATION I
      II=NOSTA(I)
      CURVE(I,1)=FM(II)
      CURVE(I,2) = C2 * (1000.0 * Y(I))**(-0.3)
 150  CONTINUE
```

PROGRAM PLOTWM

```
C       PLOT CURVES.
        IF (LP) CALL PLCURV(NS,2,Y,CURVE,0.0,0.0,0.0,0.0,
       + NCHAR,TITLE, NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
        CALL FACTOR(FAC2*FACMAIN)
        IF(PL)CALL ZPLOT(NS,2,Y,CURVE,0.0,0.0,0.0,0.0,
       +   0.0,0.0,NCHAR,TITLE(3),
       +   NXCHAR,XLABEL,NYCHAR,YLABEL,10.1,6.5,0.0,10.1,0.0,6.5)
C
        GO TO 20
C   SECTION FOR MEAN DIRECTION PLOT
  230 READ(5,1003) (TITLE(I),I=3,10,1)
      READ(5,1004) NXCHAR,XLABEL
      READ (5,1004) NYCHAR,YLABEL
      IF (SREC) GOTO 240
          CALL READS(LUN(1),BUFFER,NB,NF)
          SREC = .TRUE.
  240 IDFSET = 11 + 2*NGRID
      DO 250 I=1,NS,1
          J = NOSTA(I) + IDFSET
          CURVE(I,1) = AMOD((57.296*BUFFER(J)),360.0)
      IF (CURVE(I,1) .GT. 180.0) CURVE(I,1) = CURVE(I,1) - 360.0
  250     CONTINUE
      IF (LP) CALL PLCURV(NS,1,Y,CURVE,0.0,0.0,-180.0,180.0,
     +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
      CALL FACTOR(FAC3*FACMAIN)
      IF (PL) CALL ZPLOT(NS,1,Y,CURVE,0.0,0.0,-180.0,180.0,
     +    0.0,0.0,
     +    NCHAR,TITLE(3),NXCHAR,XLABEL,NYCHAR,YLABEL,10.1,6.5,0.0,10.1,
     +    0.0,6.5)
      GOTO 20
C   SECTION FOR 1-D SPECTRUM PLOT
  330 CONTINUE
C

      READ(5,1003) (TITLE(I),I=3,10)
      READ(5,1004) NXCHAR,XLABEL
      READ(5,1004) NYCHAR,YLABEL
      READ(5,1000) NDU
      READ(5,1005) NS1D
      NS1D = MIN0(NS1D,10)
      READ(5,1005) (NO1D(I),I=1,NS1D,1)
      WRITE(6,2330)
 2330 FORMAT(12H PLOT TYPE 4)
      WRITE(6,2331) (TITLE(I),I=3,10,1)
 2331 FORMAT(6H TITLE,/,8A10)
      WRITE(6,2332) XLABEL
 2332 FORMAT(13H X AXIS LABEL,/,6A10)
      WRITE(6,2334) YLABEL
 2334 FORMAT(13H !YAXIS LABEL,/,6A10)
      WRITE(6,2335) NS1D,(NO1D(I),I=1,NS1D,1)
 2335 FORMAT(20H NUMBER OF STATIONS:,I3,/,13HSTATION LIST:,/,(I6,3X)
      IOFSET=11
      IF (JSREC) GOTO 340
          CALL READJS(LUN(1),S1D,NF,NGRID)
          JSREC = .TRUE.
  340 DO 350 I=1,NS1D,1
      L=NO1D(I)
          DO 350 J=1,NF,1
      IF(NDU.EQ.1) GOTO 345
      CURVE(J,I)=S1D(L,J)
           GOTO 350
  345 CONTINUE
      CURVE(J,I)=S1D(L,J)/BFPM
  350         CONTINUE
  360     CONTINUE
      IF(NDU.EQ.1) GOTO 380
      DO 370 I=1,NF,1
          TAR(I)=FREQ(I)
  370     CONTINUE
      GOTO 400
  380 CONTINUE
      DO 390 I=1,NF,1
          TAR(I)=FREQ(I)*JSTAR/G/100
  390     CONTINUE
  400 CONTINUE
```

PROGRAM PLOTW4

```
      IF(LP)CALL PLCURV(NF,NS1),TAR,CURVE,0.0,4.0,0.0,1.5,
     +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
      CALL FACTOR(FAC4*FACMAIN)
      IF(PL)CALL ZPLOT(NF,NS1D,TAR,CURVE,0.0,4.0,0.0,1.5,
     +  0.0,0.0,
     + NCHAR,TITLE(3),NXCHAR,XLABEL,NYCHAR,YLABEL,
     +  8.0,6.0,0.0,8.0,0.0,6.0)
      GOTO 20
C
  430 CONTINUE
      CALL FACTOR(FAC5*FACMAIN)
      CALL GRPT(NGRID,LUN(2),ND,NF,FREQ,LP,PL,CURVE,NEWPAGE,NX,NY,
     +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL)
      GO TO 20
  530 CONTINUE
      CALL FACTOR(FACMAIN*FAC6)
      SREC=.FALSE.
      CALL S2DPLT(NGRID,LUN(2),ND,NF,FREQ,LP,PL,NEWPAGE,NX,NY,NCHAR,
     +   TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,CURVE,BUFFER)
      GO TO 20
C     LEAVE ROOM FOR OTHER PLOTS.
  630 CONTINUE
      CALL FACTOR(FAC5*FACMAIN)
      SREC=.FALSE.
      CALL S1DLT(NGRID,LUN(2),ND,NF,FREQ,LP,PL,NEWPAGE,NX,NY,NCHAR,
     +   TITLE,NXCHAR,NYCHAR,TAR,CURVE,BUFFER)
      GOTO 20
      END
```

```
      SUBROUTINE PKFREQ(FREQ,S1D,NS,NF,FM,NOSTA)
C   THIS SUBROUTINE CALCULATES THE PEAK FREQUENCY WHICH IS RETURNED IN F
C
C   FREQ - ARRAY OF FREQUENCIES
C   S1D - 1-D SPECTRUM VALUES
C   NS - NUMBER OF STATIONS
C   NF - NUMBER OF FREQUENCIES
C
C   I - STATION COUNTER
C   J - FREQUENCY INDEX
C   SM - HOLDS MAX FREQUENCY DURING SEARCH ROUTINE
C   JM - INDEX OF PEAK FREQUENCY
C   J1 - JM - 1
C   J2 - JM
C   J3 - JM + 1
C   X1 - FIRST FREQUENCY
C   Y1 - FIRST SPECTRAL VALUE
C   X2 - SECOND FREQUENCY
C   Y2 - SECOND SPECTRAL VALUE
C   X3 - THIRD FREQUENCY
C   Y3 - THIRD SPECTRAL VALUE
C   XNUM - NUMERATOR OF FREQUENCY APPROXIMATION
C   DENOM - DENOMINATOR OF FREQUENCY APPROXIMATION
C
      INTEGER NS, NF, I, J, JM, J1, J2, J3
      INTEGER NOSTA(NS)
      REAL FREQ(NS),S1D(500,20),FM(NF).SM,X1,X2,X3,Y1,Y2,Y3
      REAL XNUM, DENOM
C   DO EVERYTHING FOR EACH STATION
      DO 40 K=1,NS,1
         I=NOSTA(K)
C   BEGIN SEARCH FOR DISCRETE PEAK
C   DO PRELIMINARIES
         SM = S1D(I,1)
         JM = 1
C   START ACTUAL SEARCH
         DO 10 J=2, NF, 1
            IF (SM .GE. S1D(I,J)) GO TO 10
C   KEEP NEW MAX
               SM = S1D(I,J)
               JM = J
   10       CONTINUE
C   SEE IF CURVE FITTING CAN BE DONE
         IF ((JM .NE. 1) .AND. (JM .NE. NF)) GO TO 20
C   SINCE PEAK IS AT END, CURVE FITTING CAN NOT BE DONE
            FM(I) = FREQ(JM)
            GO TO 40
C   PREPARE TO DO CURVE FITTING BY SOLVING PEAK AS MAXIMUM VALUE OF QUAD
C    THAT PASSES THROUGH THE 3 POINTS AROUND THE DISCRETE PEAK
   20 J1=JM-1
      J2=JM
      J3=JM+1
      X1=FREQ(J1)
      Y1=S1D(I,J1)
      X2=FREQ(J2)
      Y2=S1D(I,J2)
      X3=FREQ(J3)
      Y3=S1D(I,J3)
C   CALCULATE DENOMINATOR OF PEAK FREQUENCY
      DENOM=2*(X2*Y3-X3*Y2+X3*Y1-X1*Y3+X1*Y2-X2*Y1)
C   APPROXIMATION CAN NOT BE DONE IF DENOMINATOR IS ZERO
      IF (DENOM .NE. 0) GOTO 30
         FM(I) = FREQ(JM)
         GO TO 40
C CALCULATE NUMERATOR AND PEAK FREQUENCY
   30   XNUM=-(Y2*X3*X3-Y3*X2*X2+Y3*X1*X1-Y1*X3*X3+Y1*X2*X2 -Y2*X1*X1)
      FM(I)=XNUM/DENOM
   40  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRPT(NGRID,LUN,ND,NF,FREQ,LP,PL,E,NEWPAGE,NX,NY,
     +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL)
C
C   THIS ROUTINE READS THE HISTORY TAPE AND PRODUCES A GRAPH
C    OF THE 1-D SPECTRUM AT SELECTED TIMES AT ONE GRID POINT.
C   IT CALLS:
C    FNDIDREC - POSITIONS FILE POINTER AT BEGINNING OF RIGHT DATA RECORD
C    RSPLIT - UNPACKS A REAL VALUE FROM A WORD
C    ISPLIT - UNPACKS AN INTEGER VALUE FROM A WORD
C    PLCURV - ROUTINE TO DO LINE PRINTER PLOTS
C    ZPLOT - ZETA PLOT ROUTINE
C
C  PARAMETERS
C   NGRID - TOTAL NUMBER OF GRID POINTS
C   LUN - LOGICAL UNIT NUMBER
C   ND - NUMBER OF DIRECTIONS
C   NF - NUMBER OF FREQUENCIES
C   FREQ - ARRAY OF FREQUENCIES
C   LP - FLAG FOR LINE PRINTER PLOTS
C   PL - FLAG FOR PLOTTER PLOTS
C   NCHAR - LENGTH OF TITLE
C   TITLE - TITLE OF GRAPH
C   NX - WIDTH OF PRINTER PLOT
C   NXCHAR - LENGTH OF X-AXIS LABEL
C   XLABEL - XAXIS LABEL
C   NYCHAR - HEIGHT OF LINE PRINTER PLOT
C   YLABEL - ACTUAL ARRAY CONTAINING Y-AXIS LABEL
C
C  VARIABLES:
C   NSTEP - NUMBER OF TIME STEPS
C   ITIME - ARRAY OF SELECTED TIMES
C   INGRDPT GRID POINT TO BE PLOTTED
C   BUF - INPUT BUFFER
C   EOB - END OF BUFFER (LENGTH OF INPUT RECORD)
C   TIMCTR - TIME INDEX
C   J - FREQUENCY INDEX
C   K - DIRECTION INDEX
C   FACTOR - 2*PI*DISTANCE
C   IND - TELLS WHICH HALF OF WORD GRID POINT VALUE IS PACKED IN
C   IPTR - INDEX OF BUF FOR GRID POINT VALUE
C   E - ARRAY OF INDEPENDENT VARIABLE VALUES
C
      INTEGER NGRID,LUN,ND,NF,NSTEP,ITIME(10),INGRDPT,NCHAR
      INTEGER NXCHAR,NYCHAR,EOB,TIMCTR,J,K,IND,IPTR,NX,NY
      REAL FACTOR,E(100,10),FREQ(NF),TITLE(10),XLABEL(6),YLABEL(6)
      REAL BUF(254)
      REAL BFPM,USTAR,G
      LOGICAL LP,PL
      DATA BFPM/1351000.0/,USTAR/85.5/,G/980.6/
C
C   READ IN USER SUPPLIED INFORMATION
C
      READ (5,1003) (TITLE(I),I=3,10)
 1003 FORMAT(8A10)
      READ(5,1004) NXCHAR,XLABEL
 1004 FORMAT(I10,10X6A10)
      READ (5,1004) NYCHAR,YLABEL
      READ(5,1004) NDU
      READ (5,1005) NSTEP,(ITIME(I),I=1,NSTEP,1)
 1005 FORMAT(8I10)
      READ (5,1005) INGRDPT
      WRITE(6,2000)
 2000 FORMAT(17H GRPT-PLOT TYPE 5)
      WRITE(6,2001) (TITLE(I),I=3,10,1)
 2001 FORMAT(7H TITLE:8A10)
      WRITE(6,2002) NXCHAR,XLABEL
 2002 FORMAT(9H NXCHAR=,I3,/,7HXLABEL=,6A10)
      WRITE(6,2003) NYCHAR,YLABEL
 2003 FORMAT(8H NYCHAR=,I3,/,7HYLABEL=,6A10)
      WRITE(6,2004) NSTEP,(ITIME(I),I=1,NSTEP,1)
 2004 FORMAT(22H NUMBER OF TIME STEPS:,I2,/,16H,SELECTED TIMES:,(I4,3X)
      WRITE(6,2005) INGRDPT
 2005 FORMAT(12H GRID POINT:,I4)
C   READ DATA AND DO SUMATION
      EOB = 4+(1+NGRID)/2
```

SUBROUTINE GRPT

```
      REWIND LUN
      DO 240 TIMCTR = 1,NSTEP,1
         DO 200 J=1,NF,1
            E(J,TIMCTR) = 0.0
200         CONTINUE
       REWIND LUN
            CALL FNDREC(ITIME(TIMCTR),LUN,ND,NF)
         DO 230 J=1,NF,1
            DO 220 K=1,ND,1
               BUFFER IN (LUN,1) (BUF(1),BUF(EOB))
               IF (UNIT(LUN)) 210,998,998
210            FACTOR = RSPLIT(1,BUF(4))
            IND = MOD(INGRDPT+1,2) + 1
               IPTR = (INGRDPT + 1)/2 + 4
               Z=RSPLIT(IND,BUF(IPTR))
               E(J,TIMCTR)=E(J,TIMCTR)+FACTOR*Z
220            CONTINUE
230         CONTINUE
240      CONTINUE
      IF(NDU.EQ.1)GOTO260
      DO 250 J=1,NF,1
         BUF(J)=FREQ(J)
250         CONTINUE
         GOTO 290
260   CONTINUE
      DO 280 I=1,NF,1
         BUF(I)=FREQ(I)*USTAR/G
         DO 270 J=1,NSTEP,1
            E(I,J)=E(I,J)/BFPM
270         CONTINUE
280      CONTINUE
290   CONTINUE
      IF(LP)CALL PLCURV(NF,NSTEP,BUF,E,0.0,0.02,0.0,1.5,
     +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
      NF=NF-1
      IF(PL)CALL ZPLOT(NF,NSTEP,BUF,E,0.0,0.02,0.0,1.5,
     +   0.25,0.0025,
     +   NCHAR,TITLE(3),NXCHAR,XLABEL,NYCHAR,YLABEL,
     +     8.0,6.0,0.0,8.0,0.0,6.0)
      NF=NF+1
      RETURN
998   WRITE(6,999) LUN
999   FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
      STOP
      END
```

SUBROUTINE S2DPLT

```
      SUBROUTINE S2DPLT(NGRID,LUN,ND,NF,FREQ,LP,PL,NEWPAGE,NX,NY,
     +  NCHAR,TITLE,NXCHAR,  XLABEL,NYCHAR,YLABEL,S2D,BUF)
C
C  THIS ROUTINE READS THE HISTORY TAPE AND PRODUCES A GRAPH
C     OF THE 2-D SPECTRUM AT SELECTED DIRECTIONS.
C  IT CALLS:
C   FNDIDREC - POSITIONS FILE POINTER AT BEGINNING OF RIGHT DATA RECORD
C   RSPLIT - UNPACKS A REAL VALUE FROM A WORD
C   PLCURV - ROUTINE TO DO LINE PRINTER PLOTS
C   ZPLOT - ZETA PLOT ROUTINE
C
C PARAMETERS
C  NGRID - TOTAL NUMBER OF GRID POINTS
C  LUN - LOGICAL UNIT NUMBER
C  ND - NUMBER OF DIRECTIONS
C  NF - NUMBER OF FREQUENCIES
C  FREQ - ARRAY OF FREQUENCIES
C  LP - FLAG FOR LINE PRINTER PLOTS
C  PL - FLAG FOR PLOTTER PLOTS
C  NCHAR - LENGTH OF TITLE
C  TITLE - TITLE OF GRAPH
C  NX - WIDTH OF PRINTER PLOT
C  NXCHAR - LENGTH OF X-AXIS LABEL
C  XLABEL - XAXIS LABEL
C  NYCHAR - Y-AXIS LABEL LENGTH
C  YLABEL - ACTUAL ARRAY CONTAINING Y-AXIS LABEL
C
C VARIABLES:
C  INGRDPT GRID POINT TO BE PLOTTED
C  BUF - INPUT BUFFER
C  EOB - END OF BUFFER (LENGTH OF INPUT RECORD)
C  J - FREQUENCY INDEX
C  K - DIRECTION INDEX
C  IND - TELLS WHICH HALF OF WORD GRID POINT VALUE IS PACKED
C  IPTR - INDEX OF BUF FOR GRID POINT VALUE
C
      INTEGER TITLE(10),XLABEL(6),YLABEL(6)
      INTEGER IDIR(40)
      REAL FREQ(20),S2D(100,10)
      REAL BUF(600)
      LOGICAL LP,PL
C  READ IN USER SUPPLIED INFORMATION
      READ (5,1003) (TITLE(I),I=3,10)
 1003 FORMAT(8A10)
      READ(5,1004) NXCHAR,XLABEL
 1004 FORMAT(I10,10X6A10)
      READ (5,1004) NYCHAR,YLABEL
 1005 FORMAT(8I10)
      READ (5,1005) INGRDPT
      READ(5,1005) ITSTEP
      READ(5,1005) NSD,(IDIR(I),I=1,NSD,1)
      DO 20 I=1,NSD,1
         DO 10 J=1,NF,1
            S2D(J,I)=0.0
   10       CONTINUE
   20    CONTINUE
      EOB = 4+(1+NGRID)/2
      REWIND LUN
            IND = MOD(INGRDPT+1,2) + 1
                IPTR = (INGRDPT + 1)/2 + 4
      CALL FNDREC(ITSTEP,LUN,ND,NF)
      DO 50 J=1,NF,1
         I=1
      K=1
   30 CONTINUE
              BUFFER IN (LUN,1) (BUF(1),BUF(EOB))
      IF (UNIT(LUN)) 31,998,993
   31 CONTINUE
      IF(K.NE. IDIR(I)) GOTO 33
            S2D(J,I)=RSPLIT(IND,BUF(IPTR))
            I=1+I
   33 CONTINUE
      IF(ND.EQ.K) GOTO 50
      K=1+K
      IF(I.LE.NSD) GOTO 30
```

SUBROUTINE S2DPLT

```
 39 CONTINUE
    BUFFER IN (LUN,1) (BUF(1),BUF(1))
    IF(UNIT(LUN)) 40,998,998
 40 CONTINUE
    K=1+K
    IF(K.LT.ND)GOTO 39
 50    CONTINUE
    IF(LP) CALL PLCURV(NF,NSD,FREQ,S2D,0.0,0.0,0.0,0.0,
   +    NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEHPAGE,NX,NY)
    IF(PL)CALL ZPLOT(NF,NSD,FREQ,S2D,0.0,0.0,0.0,0.0,
   +  0.0,0.0,
   +  NCHAR,TITLE(3),NXCHAR,XLABEL,NYCHAR,YLABEL,
   +  11.1,7.1,0.0,11.1,0.0,7.1)
    RETURN
998 WRITE(6,999) LUN
999 FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
    STOP
    END
```

```
      SUBROUTINE S1D1T(NGRID,LUN,ND,NF,FREQ,LP,PL,NEWPAGE,NX,NY,NCHAR,
     +  TITLE,NXCHAR,NYCHAR,TAR,E)
C
      INTEGER NSTEP,NGRDPTS,GRDPTS(10),NXCHAR,XLABEL(7),NYCHAR,YLABEL(7
      INTEGER NDU,LUN,EOB,NGRID,NF,ND
      INTEGER INDEX(10,2),I1
      LOGICAL PL,LP
      REAL TITLE(10)
      REAL FREQ(NF),TAR(100),G,USTAR,BUF(254)
      REAL E(100,2),VAR(100),T,FACTOR
      REAL BFPM
      DATA BFPM/1351000.)/
      DATA USTAR/85.5/,G/980.5/
C NSTEP - SELECTED TIME STEP
C NGRDPTS - NUMBER OF GRID POINTS SELECTED
C GRPTS - ARRAY OF SELECTED GRID POINTS
C NXCHAR - NUMBER OF CHARACTERS IN X-AXIS LABEL
C TITLE - TITLE OF GRAPH
C XLABEL - X AXIS LABEL
C NYCHAR - NUMBER OF CHARACTERS IN Y-AXIS LABEL
C YLABEL - Y-AXIS LABEL
C NDU - FLAG FOR NON-DIMENSIONAL UNITS
C LUN - LOGICAL UNIT NUMBER
C EOB END OF BUFFER
C NGRID - NUMBER OF GRID POINTS ON TAPE
C NF - NUMBER OF FREQUENCIES
C FREQ - ARRAY OF FREQUENCIES
C TAR - TEMP ARRAY, USED TO STORE INDEPENDENT VARIABLE VALUES
C G - GRAVITY CONSTANT
C USTAR - U* CONSTANT
C ND - NUMBER OF DIRECTIONS
C BUF - ARRAY USED AS INPUT BUFFER
      READ(5,1000) NSTEP
 1000 FORMAT(8I10)
      READ(5,1000) NGRDPTS
      NGRDPTS=MINO(NGRDPTS,10)
      READ(5,1000) (GRDPTS(I),I=1,NGRDPTS,1)
      READ(5,1002) NCHAR,(TITLE(I),I=3,9,1)
 1001 FORMAT(8A10)
      READ(5,1002) NXCHAR,XLABEL
 1002 FORMAT(I10,(7A10))
      READ (5,1002) NYCHAR,  YLABEL
      READ(5,1000) NDU
      WRITE(6,2000) NSTEP
 2000 FORMAT(20H SELECTED TIME STEP:,I4)
      WRITE(6,2001) NGRDPTS,(GRDPTS(I),I=1,NGRDPTS,1)
 2001 FORMAT(32H NUMBER OF SELECTED GRID POINTS:,I4,/,
     +   21H LIST OF GRID POINTS:,/,2X,10I8)
      WRITE(6,2002) (TITLE(I),I=3,10)
 2002 FORMAT(7H TITLE:,/,2X,8A10)
      WRITE(6,2003) NXCHAR,XLABEL
 2003 FORMAT(24H LENGTH OF X-AXIS LABEL:,I4,/,14H X-AXIS LABEL:,7A10)
      WRITE(6,2004) NYCHAR,YLABEL
 2004 FORMAT(24H LENGTH OF Y-AXIS LABEL:,I4,/,14H Y-AXIS LABEL:,7A10)
      WRITE(6,2005) NDU
 2005 FORMAT(5H NDU=,I2)
      REWIND LUN
      EOB=4+(1+NGRID)/2
      DO 5 I=1,NGRDPTS,1
         INDEX(I,1)=(GRDPTS(I)+1)/2 +4
         INDEX(I,2)=MOD(GRDPTS(I)+1,2)+1
    5    CONTINUE
      IF(NDU .EQ. 1) GOTO 20
         DO 10 I=1,NF,1
            TAR(I)=FREQ(I)
   10       CONTINUE
         GOTO 40
   20    CONTINUE
         DO 30 I=1,NF,1
            TAR(I)=FREQ(I)*USTAR/G
   30       CONTINUE
   40    CONTINUE
      DO 60 I=1,NF,1
         DO 50 J=1,NGRDPTS,1
            E(I,J)=0.0
```

SUBROUTINE S101T

```
   50          CONTINUE
   60       CONTINUE
      Z=0.0
      REWIND LUN
      CALL FNDREC(NSTEP,LUN,ND,NF)
      DO 100 J=1,NF,1
         DO 90 K=1,ND,1
            BUFFER IN (LUN,1) (BUF(1),BUF(EOB))
            IF(UNIT(LUN)) 70,999,999
   70          CONTINUE
            FACTOR=RSPLIT(1,BUF(4))
            DO 80 I=1,NGRDPTS,1
               I1=INDEX(I,1)
               T=RSPLIT(INDEX(I,2),BUF(I1))
               E(J,I)=E(J,I)+FACTOR*T
      Z=AMAX1(Z,E(J,I))
   80          CONTINUE
   90       CONTINUE
  100    CONTINUE
      IF(NDU .NE. 1) GOTO 130
         DO 120 I=1,NGRDPTS,1
            DO 110 J=1,NF,1
               E(J,I)=E(J,I)/BFPM
  110          CONTINUE
  120       CONTINUE
  130 CONTINUE
      DO 131 I=1,NGRDPTS,1
         DO 131 J=1,NF,1
            E(J,I)=E(J,I)/Z
  131       CONTINUE
      NF=18
      IF(LP)CALL PLCURV(NF,NGRDPTS,TAR,E,0.0,0.2,0.0,1.0,
     +  NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
      IF(PL)CALL ZPLOT(NF,NGRDPTS,TAR,E,0.0,0.2,0.0,1.0,
     +   0.16557,0.025,
     +  NCHAR,TITLE(3),NXCHAR,XLABEL,NYCHAR,YLABEL,
     +   8.0,5.0,0.0,8.0,0.0,5.0)
      NF=20
      RETURN
  999 CONTINUE
      WRITE(6,1111) LUN
 1111 FORMAT(25H ERROR WHEN READING UNIT ,I2)
      RETURN
      END
```

FUNCTION RSPLIT

```
      FUNCTION RSPLIT(IND,WORD)
C
      INTEGER IND
      REAL WORD,TEMP
      GO TO (100,110) , IND
  100 RSPLIT = WORD .AND. 77777777770000000000B
      RETURN
  110 TEMP = WORD .AND. 00000000007777777777B
      RSPLIT = SHIFT(TEMP,30)
      RETURN
      END
```

SUBROUTINE LIMITS

```
      SUBROUTINE LIMITS(XMIN,XMAX,IROW,ICOL,NPTS,X)
C
C  THIS SUBROUTINE FINDS THE MINIMUM AND MAXIMUM VALUE OF A 2-D ARRAY
C
C  XMIN - MINIMUM RETURNED HERE
C  XMAX - MAXIMUM RETURNED HERE
C  X - ARRAY OF VALUES TO BE SEARCHED
C  IROW - NUMBER OF ROWS IN ARRAY
C  ICOL - NUMBER OF COLUMNS IN ARRAY TO BE SEARCHED
C  NOTE: IF ICOL=1 THEN A 1-D ARRAY CAN BE SEARCHED.
C NPTS - NUMBER OF POINTS IN ARRAY
C
      INTEGER IROW,ICOL,NPTS
      REAL XMIN,XMAX,X(IROW,ICOL)
C  INITIALIZE MINIMUM AND MAXIMUM
      XMIN = X(1,1)
      XMAX = XMIN
C  NOW START SEARCH
      DO 10 I=1,ICOL,1
         DO 10 J=1,NPTS,1
            XMIN = AMIN1(XMIN,X(J,I))
            XMAX = AMAX1(XMAX,X(J,I))
   10 CONTINUE
      RETURN
      END
```

FUNCTION ISPLIT

```
      FUNCTION ISPLIT(IND,IWORD)
C
      INTEGER IND,IWORD,ITEMP
      GOTO (100,110),IND
  100 ITEMP = IWORD .AND. 77777777770000000000B
      ISPLIT = SHIFT(ITEMP,-30)
      RETURN
  110 ITEMP = IWORD .AND. 00000000007777777777B
      ITEMP = SHIFT(ITEMP,30)
      ISPLIT = SHIFT(ITEMP,-30)
      RETURN
      END
```

SUBROUTINE FNDREC

```
      SUBROUTINE FNDREC(NSTEP,LUN,ND,NF)
C
C  THIS SUBROUTINE SPACES TAPE FORWARD UNTIL PROPER TIME STEP IS FOUND
      INTEGER IBUF(5),NSTEP,LUN,ND,NF,I,J
C  NSTEP - TIME STEP WANTED
C  LUN - LOGICAL UNIT NUMBER
C  ND - NUMBER OF DIRECTIONS
C  NF - NUMBER OF FREQUENCIES
C  IBUF - INPUT BUFFER
C  I - DIRECTION INDEX
C  J - FREQUENCY INDEX
C
C READ IN ID RECORD
C
      BUFFER IN (LUN,1) (IBUF(1),IBUF(5))
C  MAKE SURE NO READ ERRORS
      IF (UNIT(LUN)) 10,998,998
C  PULL OUT PRESENT STEP NUMBER
   10 IBUF(1) = ISPLIT(2,IBUF(1))
C  IS IT THE RIGHT STEP NUMBER?
      IF (IBUF(1) .EQ. NSTEP) GO TO 900
C  NO, WELL SKIP ALL THE DATA RECORDS THEN
   20 DO 40 J=1,NF,1
         DO 30 K=1,ND,1
            BUFFER IN (LUN,1) (IBUF(1),IBUF(1))
C  MAKE SURE A READ ERROR DID NOT OCCUR
            IF (UNIT(LUN)) 30,998,998
   30       CONTINUE
   40    CONTINUE
C  NOW READ IN ID RECORD
      BUFFER IN (LUN,1) (IBUF(1),IBUF(5))
C  CHECK FOR READ ERROR
            IF (UNIT(LUN)) 50,998,998
C  PULL OUT CURRENT STEP NUMBER
   50 IBUF(1) = ISPLIT(2,IBUF(1))
C  IS THIS RIGHT STEP?
      IF (IBUF(1) .NE. NSTEP) GOTO 20
C MUST BE RIGHT PLACE
  900 RETURN
  998 WRITE(5,999) LUN
  999 FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
      STOP
      END
```

```
      SUBROUTINE GRIDV(XMX,XMN,YMX,YMN,DDX,DDY,NX,NY,IER)
C  ROUTINE TO BLANK OUT PLOT IMAGE, THEN PUT IN AXES
C     XMX,XMN- MAX AND MIN X AXIS VALUES
C     YMX,YMN- MAX AND MIN Y AXIS VALUES
C     DDX,DDY- X DISTANCE OR Y DISTANCE BETWEEN PRINTED AXIS VALUES
C     NX,NY- NO. PRINTER SPACES FOR X AND Y AXIS
C     IER- ERROR RETURN CODE (0 INDICATES OK)
C
      INTEGER BLANK,PLUS
      COMMON /ZZZZ/IMAG(121,61),
     +             XMAX,XMIN,YMAX,YMIN,
     +    DX,DY,NNX,NNY,
     +             XSCALE,YSCALE,
     +             MX,MY,IG,IV
      DATA BLANK,MINUS,PLUS,ICHAR/10H          ,1H-,1H+,1HI/
      IYF(YY)=MY-INT(YSCALE*(YY-YMIN)+.5)
      IXF(XX)=1.5+XSCALE*(XX-XMIN)
C
      IER=0
      IG=0
      IV=0
C
C  DO SOME ERROR CHECKING
      IF (XMX.LE.XMN) GO TO 900
      XMAX=XMX
      XMIN=XMN
      DELX=XMAX-XMIN
      IF (YMX.LE.YMN) GO TO 901
      YMAX=YMX
      YMIN=YMN
      DELY=YMAX-YMIN
      DX=DDX
      DY=DDY
C
      IF (NX.LT.5.OR.NX.GT.120) GO TO 902
      MX=NX+1
      IF (NY.LT.5.OR.NY.GT.60) GO TO 903
      MY=NY+1
C  CLEAR PRINT IMAGE
      DO 200 J=1,MY
      DO 200 I=1,MX
200   IMAG(I,J)=BLANK
C
C  DETERMINE SCALE FACTORS
      XSCALE=NX/DELX
      YSCALE=NY/DELY
C
C  CREATE HORIZONTAL AND VERTICAL BORDERS
      DO 210 I=2,MX
      IMAG(I,1)=MINUS
210   IMAG(I,MY)=MINUS
      IX=IXF(XMIN)
      IMAG(IX,1)=PLUS
      IMAG(IX,MY)=PLUS
      NNX=(DELX/DX+.001)
      DO 211 I=1,NNX
      XVAL=XMIN+I*DX
      IX=IXF(XVAL)
      IMAG(IX,1)=PLUS
211   IMAG(IX,MY)=PLUS
C
      DO 212 J=2,NY
      IMAG(1,J)=ICHAR
212   IMAG(MX,J)=ICHAR
      NNY=(DELY/DY+.001)
      DO 213 J=1,NNY
      IY=IYF(YMIN+J*DY)
      IMAG(1,IY)=PLUS
213   IMAG(MX,IY)=PLUS
C
C   IF X=0 WITHIN GRID RANGE, DRAW X AXIS
      IX=IXF(0.)
      IF (IX.LE.1 .OR. IX.GT.NX) GO TO 220
      DO 221 J=2,NY
221   IMAG(IX,J)=ICHAR
```

SUBROUTINE GRIDV

```
C   IF Y=0 WITHIN GRID RANGE, DRAW Y AXIS
220     CONTINUE
        IY=IYF(0.)
        IF(IY.LE.1 .OR. IY.GT.NY) GO TO 222
        DO 223 I=2,NX
223     IMAG(I,IY)=MINUS
C
222     IG=1
        RETURN
C
900     WRITE (6,600)
600     FORMAT (45H ** ** ** XMAX<=XMIN. ERROR RETURN FROM GRID.)
        IER=1
        RETURN
901     WRITE (6,601)
601     FORMAT (45H ** ** ** YMAX<=YMIN. ERROR RETURN FROM GRID.)
        IER=2
        RETURN
902     WRITE (6,602)
602     FORMAT (76H ** ** ** X EXTENT OF GRAPH TOO SMALL OR TOO LARGE.
       +ROR RETURN FROM GRID.)
        IER=3
        RETURN
903     WRITE (6,603)
603     FORMAT (76H ** ** ** Y EXTENT OF GRAPH TOO SMALL OR TOO LARGE.
       +ROR RETURN FROM GRID.)
        IER=4
        RETURN
        END
```

```
      SUBROUTINE PLOTV(X,Y,NPTS,ISYM,NOUT,IER)
C   ROUTINE TO PLOT VECTOR Y AGAINST VECTOR X
      DIMENSION X(1),Y(1)
      INTEGER BLANK,XCHAR
      COMMON /ZZZZ/IMAG(121,61),
     +            XMAX,XMIN,YMAX,YMIN,
     +      DX,DY,NNX,NNY,
     +            XSCALE,YSCALE,
     +            MX,MY,IG,IV
      DATA BLANK,XCHAR/1H ,1HX/
      DATA MINUS,PLUS,ICHAR/1H-,1H+,1HI/
      IXF(XX)=1.5+XSCALE*(XX-XMIN)
      IYF(YY)=MY-INT(YSCALE*(YY-YMIN)+.5)
C
      IF (IG.EQ.0) GO TO 900
      IF (NPTS.LT.1) GO TO 901
      NOUT=0
C
C   PLOT NPTS OF DATA FROM X AND Y ARRAYS
      DO 100 I=1,NPTS
C   FIND X POSITION
      IX=IXF(X(I))
      IF (IX.LT.1 .OR. IX.GT.MX) GO TO 110
C
C   FIND Y POSITION
      IY=IYF(Y(I))
      IF (IY.LT.1 .OR. IY.GT.MY) GO TO 110
      LOC=IMAG(IX,IY)
C   IS PRINT POSITION BLANK OR A -, +, OR I?
      IF (LOC.EQ.BLANK) GO TO 111
      IF (LOC.EQ.MINUS
     +.OR.LOC.EQ.PLUS
     +.OR.LOC.EQ.ICHAR) GO TO 111
C   IF SOMETHING ELSE IN THE PRINT POSITION, REPLACE WITH X
      IMAG(IX,IY)=XCHAR
      GO TO 100
C   PUT PLOTTING SYMBOL IN THIS PRINT POSITION
111   IMAG(IX,IY)=ISYM
      GO TO 100
C   COME HERE IF POINT OUTSIDE GRID
110   NOUT=NOUT+1
100   CONTINUE
C
      IV=1
      RETURN
C
900   WRITE (6,600)
600   FORMAT (41H ** ** ** ERROR IN GRIDV.  NO PLOT DRAWN.)
      IER=1
      RETURN
901   WRITE (6,601)
601   FORMAT (33H ** ** ** NPTS<1.  NO PLOT DRAWN.)
      IER=2
      RETURN
      END
```

```
      SUBROUTINE LABLV(NXCHAR,XLABEL,NYCHAR,YLABEL,NCHAR,TITLE,NEWPAGE,
     +IER)
C  ROUTINE TO PRINT COMPLETED GRAPH AND LABELS AND TITLE
      INTEGER BLANK
      LOGICAL NEWPAGE
      DIMENSION XLABEL(1),YLABEL(1),TITLE(1)
      DIMENSION XV(7),FMT(2)
      COMMON /ZZZZ/IMAG(121,61),
     +                XMAX,XMIN,YMAX,YMIN,
     +      DX,DY,NNX,NNY,
     +                XSCALE,YSCALE,
     +                MX,MY,IG,IV
      DATA MASK/00000000007777000000B/,BLANK/1H /
      DATA     FMT/10H(16X,00A1,,6H11A10)/
      IXF(XX)=1.5+XSCALE*(XX-XMIN)
      IYF(YY)=MY-INT(YSCALE*(YY-YMIN)+.5)
C
      IF (IG.EQ.0) GO TO 900
      IF (IV.EQ.0)  GO TO 901
C
C  IF REQUESTED, SKIP TO A NEW PAGE
      IF (NEWPAGE) WRITE (6,680)
680   FORMAT (1H1)
C  SKIP 3 LINES AND WRITE PLOT TITLE (100CHAR'S PER LINE, 10 LINES MAX
      IF (NCHAR.EQ.0) GO TO 10
      NCH=(NCHAR+9)/10
      WRITE (6,600) (TITLE(I),I=1,NCH)
600   FORMAT (  10(/,6X,10A10))
C
C  COMPUTE STARTING LINE OF Y AXIS LABEL
10    CONTINUE
      IP1=(NYCHAR+1)/2
      IP2=MY/2
      L1=IP2-IP1
      IF (L1.LT.1) L1=1
      L2=L1+NYCHAR
      IF (L2.GT.MY) L2=MY
C
C  PRINT PLOT IMAGE, Y AXIS VALUES, AND Y LABEL
      JY1=NNY
      YVAL=YMIN+JY1*DY
      JYV=IYF(YVAL)
      ISH=0
      ICNT=0
      IL=1
      DO 100 IY=1,MY
C  WRITE PLOT IMAGE LINE
      WRITE (6,601) (IMAG(IX,IY),IX=1,MX)
601   FORMAT (14X,121A1)
C  IF APPROPRIATE, CALCULATE AND PRINT Y AXIS VALUE
      IF (JYV.GT.IY) GO TO 101
C  PRINT A Y AXIS VALUE ON THIS LINE
      WRITE (6,602) YVAL
602   FORMAT (1H+,2X,G10.4)
      JY1=JY1-1
      YVAL=YMIN+JY1*DY
      JYV=IYF(YVAL)
C  IF APPROPRIATE, WRITE A CHARACTER OF THE Y LABEL
101   IF (NYCHAR.EQ.0) GO TO 100
      IF (IY.LT.L1 .OR. IY.GT.L2) GO TO 100
      ICNT=ICNT+1
      IF (ICNT.GT.NYCHAR) GO TO 100
      YL=SHIFT(YLABEL(IL),ISH)
      ISH=ISH+6
      IF (ISH.LE.54) GO TO 102
      ISH=0
      IL=IL+1
102   NCNT=NCNT+1
      WRITE (6,603) YL
603   FORMAT (1H+,1A1)
C
100   CONTINUE
C
C  CALCULATE AND PRINT X AXIS VALUES
      WRITE (6,604) XMIN
```

SUBROUTINE LABLV

```
604    FORMAT (T10,G11.4)
       IPNT=21
       DO 200 I=1,NNX
       XX=XMIN+I*DX
       IT=IXF(XX)+9
       IF (IT.LT.IPNT) GO TO 200
       IF (IT.GT.124) GO TO 200
       IPNT=IT+11
       WRITE (6,605) IT,XX
605    FORMAT (1H+,T=,G11.4)
200    CONTINUE
C
C   WRITE X LABEL
       IF (NXCHAR.EQ.0) RETURN
       IP1=NXCHAR/2
       IP2=MX/2
       L1=IP2-IP1
       IF (L1.LT.1) L1=1
       L2=NXCHAR
       IF (L2.GT.(MX-L1)) L2=MX-L1
       NSKP=12+L1
       L2=(L2+9)/10
       WRITE (6,670) NSKP,(XLABEL(I),I=1,L2)
670    FORMAT (/,=X,11A10)
       RETURN
C
900    WRITE (6,690)
690    FORMAT (45H ** ** ** ERROR IN GRID. NO PLOT CONSTRUCTED.)
       IER=1
       RETURN
901    WRITE (6,691)
691    FORMAT (41H ** ** ** ERROR IN PLOTV.  NO PLOT DRAWN.)
       IER=2
       RETURN
       END
```

SUBROUTINE PLTRND

```
      SUBROUTINE PLTRND (A, B, X, Y)
C
C     SUBROUTINE TO FIND "NICE" PLOT BOUNDARIES FOR INTERVAL (A, B)
C
C     A = INPUT INTERVAL LOWER MARGIN.
C     B = INPUT INTERVAL UPPER MARGIN.
C     X = LOWER PLOT BOUNDARY FOR THE INTERVAL.
C     Y = UPPER PLOT BOUNDARY FOR THE INTERVAL.
C
      Q(P) = AINT (ABS (P) * Z)
C
      IF (ABS (A) .NE. ABS (B)) GO TO 5
      X = A
      Y = B
C
      RETURN
C     WITH A & B DECIMAL POINTS ALIGNED FIND HIGHEST-ORDER
C         UNEQUAL DIGIT.
 5    Z = 1.0
      IF (ABS (A) .GE. Z  .OR.  ABS (B) .GE. Z) GO TO 20
 10   IF (Q(A) .NE. Q(B)) GO TO 22
      Z = Z * 10.0
      GO TO 10
C
 20   IF (Q(A) .EQ. Q(B)) GO TO 24
 22   Z = Z / 10.
      GO TO 20
 24   Z = Z * 10.
C     IF ONE OF THE DIGITS IS 0 MOVE ONE PLACE RIGHT.
      IF (Q(A) .NE. 0.  .AND.  Q(B) .NE. 0) GO TO 32
 30   Z = Z * 10.0
C
 32   X = AINT (A * Z) / Z
      IF (A .LT. 0.0  .AND.  A .NE. X) X = AINT (A * Z  -  1.0) / Z
C
      Y = AINT (B * Z) / Z
      IF (B .GT. 0.0  .AND.  B .NE. Y) Y = AINT (B * Z  +  1.0) / Z
      IF ((X - Y) / (A - B) .GT. 2.0) GO TO 30
C
      RETURN
C
      END
```

SUBROUTINE READS

```
      SUBROUTINE READS (LUN, BUFFER, NB, NF)
C
C     SUBROUTINE TO READ SUMMARY RECORD - -
C          SUMS OVER FREQUENCY AND DIRECTION.
C
      REAL BUFFER(NB)
C
      REWIND LUN
C     SKIP TO SREC.
      DO 10 I=1,NF
      BUFFER IN (LUN,1) (DUM,DUM)
C
      IF (UNIT (LUN)) 10, 50, 50
C
 10   CONTINUE
C     READ SREC.
      BUFFER IN (LUN,1) (BUFFER(1),BUFFER(NB))
C
      IF (UNIT (LUN)) 20, 50, 50
C
 20   CONTINUE
      REWIND LUN
C
      RETURN
C
 50   WRITE (6,1000) LUN
 1000 FORMAT (32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
C
      STOP
C
      END
```

FUNCTION XINCR

```
      FUNCTION XINCR (X1, X2)
C
C     FIND "NEAT" TIC INTERVALS.
C     ADAPTED FROM TEKTRONIX PLOT 50 SOFTWARE.
C
      DATA XNOINT /10./
C
      RAW = (X2 - X1) / XNOINT
      IEXP = ALOG10 (RAW)
      S = 10.**IEXP
      IF (RAW .LT. 1.  .AND.  S .NE. RAW) S = 0.1 * S
      T = RAW / S
      IF (T .GT. 2.) GO TO 10
      IF (ABS (T - 1.) .GT. .01) S = 2. * S
      GO TO 30
   10 IF (T .GT. 5.) GO TO 20
      S = 5. * S
      GO TO 30
   20 S = 10. * S
C     S = "NEAT" TIC INTERVAL.
C     ADJUST DATA MINIMUM.
   30 M1 = X1 / S
      XM1 = S * FLOAT (M1 + 2)
   40 IF (XM1 .LE. X1) GO TO 50
      XM1 = XM1 - S
      GO TO 40
C     ADJUST DATA MAXIMUM.
   50 M2 = X2 / S
      XM2 = S * FLOAT (M2 - 2)
   60 IF (X2 .LE. XM2) GO TO 70
      XM2 = XM2 + S
      GO TO 60
C     SET VALUES TO RETURN TO CALLING PROGRAM.
   70 X1 = XM1
      X2 = XM2
      XINCR = S
C
      RETURN
C
      END
```

SUBROUTINE READJS

```
      SUBROUTINE READJS(LUN,S1D,NF,NGRID)
C
C     THIS SUBROUTINE READS 1-D SPECTRAL VALUES FOR SELECTED STATIONS
C
C     S1D - RETURNS RESULT OF SUBROUTINE
C     BUFFER - TEMPORARY INPUT BUFFER
C     LUN - LOGICAL UNIT NUMBER
C     J - FREQUENCY DO LOOP INDEX
C     L - BUFFER INDEX
C
      REAL BUFFER(501), S1D(500,NF)
      INTEGER NF,LUN,I,J,NGRID
C
C     SET FILE POINTER
      REWIND LUN
C     DO UNTIL ALL FREQUENCIES HAVE BEEN PROCESSED
      DO 30 J=1, NF, 1
         BUFFER IN (LUN,1) (BUFFER(1), BUFFER(501))
         IF (UNIT(LUN)) 10,40,40
10          DO 20 I=1,NGRID,1
               S1D(I,J)=BUFFER(I+1)
20             CONTINUE
30          CONTINUE
C     RESET FILE POINTER
      REWIND LUN
      RETURN
C     ERROR CONDITION
40    WRITE(6,50) LUN
50    FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
      STOP
      END
```

```
      SUBROUTINE P_CURV(NPTS,NGRPH,X,Y,XMN,XMX,YMN,YMX,
     +     NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,NEWPAGE,NX,NY)
C   THIS SUBROUTINE DOES PRINTER PLOTS OF UP TO 10 CURVES WITH UP
C     TO 100 POINTS EACH.
C
C   NPTS - ACTUAL NUMBER OF POINTS PER CURVE
C   NGRPH - NUMBER OF CURVES PER GRAPH
C   X - X VALUES TO BE PLOTTED
C   Y - EACH COLUMN IS A SET OF Y VALUES
C   THE NEXT FOUR VARIABLES ARE PASSED PARAMETERS THAT MAY BE CONSTANTS.
C   IN PARENTHESES ARE THE VARIABLE NAMES THAT THIS ROUTINE WILL USE FOR
C   CALCULATIONS.
C   XMN(XMIN) - MINIMUM X VALUE
C   XMX(XMAX) - MAXIMUM X VALUE
C   NOTE: IF XMN=XMX THENA SEARCH IS PERFORMED TO FIND X LIMITS
C   YMN(YMIN) - MINIMUM Y VALUE
C   YMX(YMAX) - MAXIMUM Y VALUE
C   NOTE: IF YMN=YMX THEN A SEARCH IS PERFORMED TO FIND Y LIMITS
C   NCHAR - NUMBER OF CHARACTERS IN TITLE
C   TITLE - ACTUAL TITLE IN CHARACTERS
C   NXCHAR - NUMBER OF CHARACTERS IN X-AXIS LABEL
C   XLABEL - X-AXIS LABEL
C   NYCHAR - NUMBER OF CHARACTERS IN Y-AXIS LABEL
C   YLABEL - YAXIS LABEL
C   NEWPAGE - FLAG FOR WRITING A TOP-OF-PAGE
C   NX - NUMBER OF PRINTER SPACES FOR X-AXIS
C   NY - NUMBER OF PRINTER SPACES FOR Y-AXIS
C
C   DDX - DISTANCE BETWEEN X TIC MARKS
C   DDY - DISTANCE BETWEEN Y TIC MARKS
C   ISYM - CHARACTERS USED TO PLOT CURVES
C   IER - ERROR CODE  (0 IS OK)
C   NOUT - NUMBER OF POINTS OUTSIDE OF GRID
C
      LOGICAL NEWPAGE
      INTEGER NPTS,NGRPH,NCHAR,NXCHAR,NYCHAR,NX,NY,ISYM(10),IER,NOUT
      REAL X(100),Y(100,10),XMN,XMIN,XMX,XMAX,YMN,YMIN,YMX,YMAX
      REAL TITLE(10),XLABEL(1),YLABEL(1),DDX,DDY
      DATA ISYM /1H*,1H+,1H#,1H%,1H.,1H$,1HA,1HB,1HC,1HD/
C   CALCULATE LIMITS
C   DID USER SPECIFY DOMAIN?
      IF (XMN .EQ. XMX) GO TO 2
         XMIN = XMN
           XMAX=XMX
         GO TO 4
C   NOT USER SPECIFIED SO CALCULATE
    2 CALL LIMITS(XMIN,XMAX,NPTS,1,NPTS,X)
C   DID USER SPECIFY RANGE?
    4 IF(YMN .EQ. YMX) GO TO 6
         YMIN = YMN
         YMAX = YMX
         GO TO 8
C   NOT USER SPECIFIED SO CALCULATE
    6 CALL LIMITS(YMIN,YMAX,100,NGRPH,NPTS,Y)
C   END OF GETTING LIMITS
    8 CONTINUE
C   FIND TICK MARK INCREMENTS
      DDX = XINCR(XMIN,XMAX)
      DDY = XINCR(YMIN,YMAX)
C   GENERATE AXIS
      CALL GRIDV(XMAX,XMIN,YMAX,YMIN,DDX,DDY,NX,NY,IER)
C   PLOT CURVES
      DO 10 I = 1,NGRPH,1
         CALL PLOTV(X,Y(1,I),NPTS,ISYM(I),NOUT,IER)
   10 CONTINUE
C   LABEL GRAPH AND DUMP GRID
      CALL LAB_V(NXCHAR,XLABEL,NYCHAR,YLABEL,NCHAR,TITLE,NEWPAGE,IER)
      RETURN
      END
```

```
      SUBROUTINE ZPLOT(NPTS,NGRPH,X,Y,XMN,XMX,YMN,YMX,
     +   YINC,XINC,
     +   NCHAR,TITLE,NXCHAR,XLABEL,NYCHAR,YLABEL,XAXLN,YAXLN,X0,X1,Y0,Y1
C
C  NPTS - NUMBER OF POINTS
C  NGRPH - NUMBER OF CURVES TO BE GRAPHED
C  X - ARRAY OF X COORDINATES
C  Y - Y COORDINATES, EACH COLUMN IS A SET
C  NCHAR - LENGTH OF TITLE
C  TITLE - TITLE OF GRAPH
C NXCHAR - NUMBER OF CHARACTERS IN X-AXIS LABEL
C  XLABEL - ARRAY OF CHARACTERS, ACTUAL LABEL OF X-AXIS
C  NYCHAR - LENGTH OF Y-AXIS LABEL
C  YLABEL - LABEL OF Y-AXIS
C
C  THE NEXT FOUR VARIABLES ARE PASSED PARAMETERS THAT MAY BE CONSTANTS.
C  IN PARENTHESES ARE THE VARIABLE NAMES THAT THIS ROUTINE WILL USE FOR
C  CALCULATIONS.
C  XMN(XMIN) - MINIMUM X VALUE
C  XMX(XMAX) - MAXIMUM X VALUE
C  NOTE: IF XMN=XMX THENA SEARCH IS PERFORMED TO FIND X LIMITS
C  YMN(YMIN) - MINIMUM Y VALUE
C  YMX(YMAX) - MAXIMUM Y VALUE
C  NOTE: IF YMN=YMX THEN A SEARCH IS PERFORMED TO FIND Y LIMITS
C
C  XORG - INITIAL X VALUE FOR X-AXIS
C  XINC - TIC MARK INCREMENT FOR X-AXIS
C  YORG - INITIAL Y VALUE FOR Y-AXIS
C  YINC - TIC MARK INCREMENT FOR Y-AXIS
C  T - ARRAY USED FOR SCALE CALLS
C  TX - ARRAY FOR LINE CALLS; X VALUES,XORG,XINC
C  TY - ARRAY USED FOR LINE CALLS; Y VALUES,YORG,YINC
C  X0 - LEFT X COORDINATE OF GRAPH IN INCES
C  X1 - RIGHT X COORDINATE OF GRAPH IN INCHES
C  Y0 - BOTTOM Y COORDINATE OF GRAPH IN INCHES
C  Y1 - TOP Y COORDINATE OF GRAPH IN INCHES
C  T1 - X OFFSET FOR TITLE
C  T2 - Y OFFSET FOR TITLE
C  ILET - INTEGER PASSED TO LINE TO SPECIFY PLOTTING CHARACTER
C
C
      INTEGER NPTS,NGRPH,NCHAR,NXCHAR,NYCHAR
      REAL XMN,XMIN,XMX,XMAX,YMN,YMIN,YMX,YMAX,XAXLN,YAXLN,X0,X1,Y0,Y1
      REAL XORG,XINC,YORG,YINC,T1,T2
      REAL X(1),Y(100,10), TITLE(1)
      REAL XLABEL(1), YLABEL(1)
      REAL T(4), TX(102), TY(102)
      NCHAR = NCHAR - 20
C
C  FIND EXTREME VALUES IN DATA
C
C  DID USER SPECIFY DOMAIN?
      IF (XMN .EQ. XMX) GO TO 2
         XMIN = XMN
         XMAX=XMX
         GO TO 4
C  NOT USER SPECIFIED SO CALCULATE
   2 CALL LIMITS(XMIN,XMAX,NPTS,1,NPTS,X)
C  DID USER SPECIFY RANGE?
   4 IF(YMN .EQ. YMX) GO TO 6
         YMIN = YMN
         YMAX = YMX
         GO TO 8
C  NOT USER SPECIFIED SO CALCULATE
   6 CALL LIMITS(YMIN,YMAX,100,NGRPH,NPTS,Y)
C  END OF GETTING LIMITS
   8 CONTINUE
C  PREPARE PLOT ROUTINES
      CALL PLOT(1.25,0.5,-3)
C  SCALE VALUES FOR PLOT ROUTINES
      XORG=XMIN
      YORG=YMIN
      IF(XINC.NE.0.0) GOTO 10
      T(1) = XMIN
      T(2) = XMAX
```

SUBROUTINE ZPLOT

```
      CALL SCALE(T,XAXLN,2,1)
      XORG = T(3)
      XINC = T(4)
   10 CONTINUE
      IF(YINC.NE.0.0) GOTO 12
      T(1) = YMIN
      T(2) = YMAX
      CALL SCALE(T,YAXLN,2,1)
      YORG = T(3)
      YINC = T(4)
C   DRAW X-AXIS
   12 CONTINUE
      CALL AXIS(X0,Y0,XLABEL,(-NXCHAR),XAXLN,0.0,XORG,XINC)
C   DRAW YAXIS
      CALL AXIS(X0,Y0,YLABEL,NYCHAR,YAXLN,90.0,YORG,YINC)
C   DRAW TOP OF GRID
      CALL AXIS(X0,Y1,TITLE,NCHAR,XAXLN,0.0,999.0,1.0)
C   DRAW END OF GRID
      CALL AXIS(X1,Y0,0,0,YAXLN,90.0,999.0,1.0)
      T1=X0+(X1-X0)/2.0 - (NCHAR/14.0)
      T2=0.35+Y1
      CALL SYMBOL(T1,T2,0.0,TITLE,0.0,NCHAR)
      ILET = 0
      DO 20 I = 1, NPTS, 1
         TX(I) = X(I)
   20    CONTINUE
      TX(NPTS+1) = XORG
      TX(NPTS+2) = XINC
      TY(NPTS+1) = YORG
      TY(NPTS+2) = YINC
      DO 40 I = 1, NGRPH, 1
         DO 30 J = 1, NPTS, 1
            TY(J) = Y(J,I)
   30       CONTINUE
      CALL PLOT(X0,Y0,3)
         CALL LINE(TX,TY,NPTS,1,1,ILET)
      Y1=Y1-0.15
      CALL SYMBOL(X1+0.25,Y1,0.1,ILET,0.0,-1)
         ILET = ILET + 1
   40    CONTINUE
      CALL PLOT((X0+14.0),(Y0-0.5),-3)
      NCHAR = NCHAR + 20
      RETURN
      END
```

C. MEAN DIRECTION CONTOUR

This plot is a contour graph of mean direction over the grid that the wave model operated on. The X and Y axes correspond to X and Y distance on the grid. The contours are changes in the mean direction of the wave motion. This program reads from the summary tape, hence only the final time step can be plotted. Provision for a title is made. This contour, like the others, makes use of the subroutine CONTUN.

INPUT DATA LIST FOR MEAN DIRECTION PLOT

FORMAT	VARIABLE	DESCRIPTION
I10,7A10	NCHAR,TITLE	NCHAR is the number of characters in the title, TITLE is the title of the plot
I10,I10	NB,NF	NB is number of words of data to read in from tape, NF is the number of frequencies used in model run
I10	IOFSET	Number of points of data to skip, used to bypass header info or special points
2I10	IGRDX,IGRDY	Number of points in grid in X and Y directions, respectively
I10,3F10.0	MODE,AR(1), AR(2),AR(3)	Correspond to mode and AR of CONTUN, refer to CONTUN
I10,F10.0	NDGTS,SIZE	Number of decimal places to be printed, size in fractions of an inch
I10.F10.0	NONO,THETA	If NONO is negative then plots are not labeled, else plots are labeled, labeling occurs at an angle equal to THETA
2F10.0	SI,SJ	Points per inch in Z direction and Y direction, respectively

VARIABLE LIST FOR MEAN DIRECTION

AR(3)	-	Real, interval values for CONTUN
F(20,20)	-	Real, array of values to be contoured
FSMSK(530)	-	Real, mask array for CONTUN
BUFFER(2000)	-	Input buffer for READS
TITLE(7)	-	Integer, title of plot
FS(18)	-	Integer mask array for CONTUN
NCHAR	-	Number of characters for title
NB	-	Integer, size of buffer
NF	-	Integer, number of frequencies
IOFSET	-	Integer number of words to skip
NGRID	-	Integer, number of grid points
IGRDX	-	Integer, number of points in X direction
IGRDY	-	Integer, number of points in Y direction
MODE	-	Integer, indicates mode for CONTUN
NDGT	-	Integer number of decimal places to be used on contour labels
SIZE	-	Real, size of contour labels in parts of an inch
NONO	-	Integer, if negative contours are not labeled, if positive or 0 contours are labeled
THETA	-	Real, angle of rotation for contour labels
SI	-	Real, number of points per inch in X direction
SJ	-	Real, number of points per inch in Y direction
IND	-	Integer, index into BUFFER
XMAX	-	Real, length of plot in X direction in inches
YMAX	-	Real, length of box in Y direction in inches
XINDT	-	Real, how far from box edge to first point in X direction
YINDT	-	Real, how far from box bottom to first point in Y direction

```
      PROGRAM MN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE1=1)
C
C AR - INTERVAL VALUES FOR /CONTUN/
C F - INPUT ARRAY FOR /CONTUN/
C FS - MASK ARRAY
C FSMSK - MASK ARRAY FOR CONTUN
C IGRDX - NUMBER OF GRID POINTS IN X DIRECTION
C IGRDY - NUMBER OF GRID POINTS IN Y DIRECTION
C MODE - SPECIFIES MODE FOR /CONTUN/
C NONO - FLAG TO TELL /CONTUN/ WHETHER OR NOT TO LABEL CONTOURS
C NDGT - NUMBER OF DIGITS IN CONTOUR LABEL
C NPTSK - NUMBER OF GRID POINTS TO SKIP
C SIZE - HEIGHT OF CONTOUR LABELS AS PART OF AN INCH
C SI - NUMBER OF POINTS PER INCH IN X DIRECTION
C SJ - NUMBER OF POINTS PER INCH IN Y DIRECTION
C THETA - ANGLE THAT CONTOUR LABELS ARE ROTATED
C
      REAL F(20,20),FSMSK(530),AR(3),BUFFER(2000)
      INTEGER TITLE(7),FS(18)
C READ IN LENGTH OF TITLE AND TITLE OF GRAPH
      READ(5,1000) NCHAR,(TITLE(I),I=1,7,1)
C     GET BUFFER SIZE AND NUMBER OF FREQUENCIES
      READ(5,1010) NB,NF
C     GET OFFSET
      READ(5,1010) IOFSET
C   GET NUMBER OF GRID POINTS
      READ(5,1010) NGRID
C     GET X AND Y DIMENSIONS OF GRID
      READ(5,1010) IGRDX,IGRDY
C     START GETTING /CONTUN/ INFORMATION
C     GET MODE AND AR VALUES
      READ(5,1020) MODE,AR(1),AR(2),AR(3)
C     GET NUMBER OF DIGITS AND SIZE OF LABELS
      READ(5,1020) NDGT,SIZE
C GET LABELING FLAG AND ANGLE OF ROTATION FOR LABELS
      READ(5,1020) NONO,THETA
      READ(5,1030) SI,SJ
 1000 FORMAT(I10,7A10)
 1010 FORMAT(8I10)
 1020 FORMAT(I10,7F10.0)
 1030 FORMAT(8F10.0)
C     LOAD BUFFER
      CALL READS(1,BUFFER,NB,NF)
C     ZERO OUT MASK ARRAYS
      DO 98 I=1,18,1
         FS(I)=0
         FSMSK(I)=0.0
   98    CONTINUE
      DO 99 I=19,530,1
         FSMSK(I)=0.0
   99 CONTINUE
      IND=IOFSET+2*NGRID+1
      DO 20 J=1,IGRDY,1
         DO 10 I=1,IGRDX,1
            F(I,J)=AMOD((57.296*BUFFER(IND)),360.0)
            IF(F(I,J).GT.180.0)F(I,J)=F(I,J)-360.0
            IND=IND+1
   10       CONTINUE
   20    CONTINUE
C     SET UP FOR PLOT ROUTINES
C     GET DIMENSIONS OF PLOT
      XMAX=IGRDX/SI
      YMAX=IGRDY/SJ
C     INITIALIZE THE PLOTTER
      CALL PLOTS(0,0,7)
C     MOVE PEN AWAY FROM EDGE OF
      CALL PLOT(0.5,0.5,-3)
C     DRAW BOX AROUND PLOT
      CALL PLOT(0.0,YMAX,2)
      CALL PLOT(XMAX,YMAX,2)
      CALL PLOT(XMAX,0.0,2)
      CALL PLOT(0.0,0.0,2)
      XINDT=1.0/(2.0*SI)
      YINDT=1.0/(2.0*SJ)
C     SET UP NEW ORIGIN
```

PROGRAM MN

```
      CALL PLOT(XINDT,YINDT,-3)
C     PUT TITLE ON GRAPH
      CALL SYMBOL(0.0,YMAX,0.25,TITLE,0.0,NCHAR)
C     DO CONTOUR PLOT           FINALLY]]]]]]
      CALL CONTUN(F,IGRDX,IGRDY,FS,AR,MODE,SI,SJ,THETA,NOMO,SIZE,
     +      FSMSK,NDGT)
      CALL PLOT(0,0,999)
C     DUMP DATA
      WRITE(6,302) IGRDX,IGRDY
  302 FORMAT(9H GRID IS ,I4,3H X ,I4)
      WRITE(6,220) THETA,NDGT,SIZE
  220 FORMAT(7H THETA=,F10.6,5X,5HNDGT=,I4,5X,5HSIZE=,F10.6)
      WRITE(6,230) XMAX,YMAX
  230 FORMAT(13H SIZE OF BOX:,F10.6,3H X ,F10.6)
      STOP
      END
```

SUBROUTINE READS

```
      SUBROUTINE READS (LUN, BUFFER, NB, NF)
C
C     SUBROUTINE TO READ SUMMARY RECORD - -
C          SUMS OVER FREQUENCY AND DIRECTION.
C
      REAL BUFFER(NB)
C
      REWIND LUN
C     SKIP TO SREC.
      DO 10 I=1,NF
      BUFFER IN (LUN,1) (DUM,DUM)
C
      IF (UNIT (LUN)) 10, 50, 50
C
 10   CONTINUE
C     READ SREC.
      BUFFER IN (LUN,1) (BUFFER(1),BUFFER(NB))
C
      IF (UNIT (LUN)) 20, 50, 50
C
 20   CONTINUE
      REWIND LUN
C
      RETURN
C
 50   WRITE (6,1000) LUN
 1000 FORMAT (32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
C
      STOP
C
      END
```

```
      SUBROUTINE CONTUN(F,IMAX,JMAX,FS,AR,MODE,SCALEI,SCALEJ,THETA,NONO,
     1SIZE,FSMSK,NDGT)
C
C       TRACES CONTOURS THROUGH ARRAY F(IMAX,JMAX) AND PLOTS THEM
C                WHITH SCALEI,SCALEJ POINTS PR. CM. THE CONTOURS ARE
C                LABELED AT AN ANGLE THETA WITH THE X-AXES PROVIDED
C                NONO =AN INTEGER .GE.0. IF NONO.LT.0 THE CONTOURS ARE
C                NOT LABELED. SIZE IS THE SIZE OF THE LABEL.
C
C         MODE = 1 FAMILY OF CONTOURS IS TRACED WITH INTERVAL=AR(1).
C         MODE = 2 FAMILY OF APPROXIMATELY AR(1)  CONTOURS IS TRACED WITH
C                  A REASONABLY COMPUTED CONTOUR INTERVAL.
C         MODE = 3 A SINGLE CONTOUR WITH VALUE AR(1) IS TRACED.
C         MODE = 4 AR(1)=INTERVAL,AR(2)=MIN,AR(3)=MAX
C         MODE = 5 AR(1)=NR OF CONTOURS, AR(2)*FMAX=MIN,AR(3)*FMAX=MAX
C
C         IMAX COINCIDES WITH THE PAPER X-AXES. DIMENSION OF ARRAY IN X DIR
C         JMAX COINCIDES WITH THE PAPER Y-AXES. DIMENSION OF ARRAY IN Y DIR
C
C         FS   AN INTEGER ARRAYS MUST BE GIVEN DIMENSIONS 1+IMAX*JMAX/31 IN
C                CALLING PROGRAM.
C
C         NDGT=SIGNIFICANCE,IF NEG. IS COMPUTED.
C 2 S COMPLEMENT ARITMETIC
C EXTERNAL FUNTION LAND I,J$ IS THE BOOLEAN  I.E.LOGICAL$ AND OF TWO
C                FULLWORD INTEGERS.
C EXTERNAL FUNCTION LOR I,J$ IS THE BOOLEAN OR OF TWO FULLWORD INTEGERS.
C
C SPECIFICATION STATEMENTS.
C
C      CHANGE KJS 5.1.73
      INTEGER           FS(1)
      REAL JAY,JAYZRO,F(IMAX,JMAX)
      DIMENSION MASK(31),NPT(4,3),AR(3),AC(5),ALG(5)
      DIMENSION IBEG(100),JBEG(100),CONSV(100),NINSV(100)
      INTEGER FSMSK(1)
      LOGICAL HALT,LPLOT,KLOG
      LOGICAL KLING,THING
      DATA NPT/3,4,1,2,2,1,4,3,4,3,2,1/,IFLAG/0/
      DATA AC/1.25,2.,2.5,5.,10./
      DATA ALG/.097,.3,.398,.7,1.0/
C FUNCTION DEFINITION.
      G(I,A,B)=FLOAT(I)-A/(B-A)
C SET PLOTTING PARAMETERS.
      IDFS=JMAX*IMAX
      IDFS=IDFS/31+1
      ARG2=AR(1)
      KBEG=1
      KBEGMX=0
      CSALEI=1./SCALEI
      CSALEJ=1./SCALEJ
C     IF (IOUT.NE.1) GO TO 604
C     DO 602 I=1,IMAX
C     WRITE (6,603) (F(I,J),J=1,JMAX)
C 602 CONTINUE
C 603 FORMAT(1H ,21F6.0)
C 604 CONTINUE
      IF (IFLAG.EQ.5) GO TO 4
      DO 3 I=1,31
    3 MASK(I)=2**(31-I)
C
C SET UP LOOP CONTROLLING SELECTION OF CONTOURS
C
      IFLAG=5
    4 IMAX1=IMAX-1
      JMAX1=JMAX-1
      IF (MODE.NE.3) GO TO 10
      FMIN=ARG2
      CONINT=ARG2
      NCONS=1
      GO TO 15
   10 FMAX=F(1,1)
      FMIN=FMAX
      DO 12 I=1,IMAX
      DO 12 J=1,JMAX
```

```
      FMIN=AMIN1(FMIN,F(I,J))
   12 FMAX=AMAX1(FMAX,F(I,J))
      IF (MODE.NE.5) GO TO 120
      AR(3)=AR(3)*FMAX
      AR(2)=AR(2)*FMAX
      ARG2=(AR(3)-AR(2))/AR(1)
  120 CONINT=ARG2
      IF (MODE.NE.2) GO TO 13
C IF MODE =2 SELECT A CONTOUR INTERVAL
      ALGCNT=ALOG10((FMAX-FMIN)/ARG2)
      N=ALGCNT
      ALGCNT=AMOD(ALGCNT,1.)
      IF (ALGCNT.GE.0.) GO TO 18
      N=N-1
      ALGCNT=1.+ALGCNT
   18 B=ALGCNT
      A=1.
      DO 19 I=1,5
      C=ABS(ALGCNT-ALG(I))
      IF (C.GE.B) GO TO 19
      B=C
      A=AC(I)
   19 CONTINUE
      CONINT=A*10.**N
C DETERMINE NUMBER OF CONTOURS AND MINIMUM VALUE
   13 IF (FMIN.GT.0.) FMIN=FMIN+CONINT
      FMIN=CONINT*AINT(FMIN/CONINT)
      IF (FMAX.LT.0.) FMAX=FMAX-CONINT
      FMAX=CONINT*AINT(FMAX/CONINT)
      IF (MODE.LT.4) GO TO 11
      FMIN=AMAX1(FMIN,AR(2))
      FMAX=AMIN1(FMAX,AR(3))
   11 NCONS=(FMAX-FMIN)/CONINT+1.
      NCONS=MINO(NCONS,40)
      IF(NCONS.LE.0) GO TO 7000
C DETERMINE NUMBER OF DIGITS IN LABEL
   15 DEL=1.E-4*CONINT
      IF(NDGT.GE.0) GO TO 20
      NDGT=0
   21 IF (ABS(AMOD(CONINT*10.**NDGT,1.)).LT.DEL) GO TO 20
      NDGT=NDGT+1
      IF (NDGT.LT.11) GO TO 21
C INSURE THAT NO POINT IS EXACTLY ON A CONTOUR
   20 DO 32 I=1,IMAX
      DO 32 J=1,JMAX
   32 IF (ABS(AMOD(F(I,J),CONINT)).LT.DEL) F(I,J)=1.000001*F(I,J)+2.*DEL
C START CONTOUR PLOTTING
      KONTUR=0
   16 KONTUR=KONTUR+1
      CON=FMIN+CONINT*FLOAT(KONTUR-1)
C     WRITE (6,620) CON,FMIN,CONINT,KONTUR
C 620 FORMAT(1H ,5X, CON ,F10.4, FMIN ,F10.4, CONINT ,F10.4, KONT
C    1UR ,I8)
      DO 14 MP=1,IDFS
   14 FS(MP)=0
C
C BEGIN EDGE SEARCH
C
      LPLOT=.FALSE.
      JE=0
   17 JE=JE+1
      IRET=1
   42 D=F(IMAX,JE)-CON
      C=F(IMAX,JE+1)-CON
      IF (C*D.GT.0.) GO TO 22
      IE=IMAX1
      NIN=3
      EYE=FLOAT(IMAX)
      JAY=G(JE,D,C)
      GO TO 50
   22 IF (JE.NE.JMAX1) GO TO 17
      IE=0
   23 IE=IE+1
      IRET=2
   43 A=F(IE,1)-CON
```

```
      D=F(IE+1,1)-CON
      IF (A*D.GT.0.) GO TO 24
      NIN=4
      JE=1
      EYE=G(IE,A,D)
      JAY=1.
      GO TO 50
   24 IRET=3
   41 B=F(IE,JMAX)-CON
      C=F(IE+1,JMAX)-CON
      IF(B*C.GT.0.)GO TO 26
      NIN=2
      JE=JMAX1
      EYE=G(IE,B,C)
      JAY =FLOAT(JMAX)
      GO TO 50
   26 IF(IE.NE.IMAX1)GO TO 23
C  BEGIN INTERIOR SEARCH
      IE=0
   27 IE=IE+1
      LPLOT=.TRUE.
      IF(IE.EQ.1)LPLOT=.FALSE.
      JE=0
   39 JE=JE+1
      IRET=4
   40 A=F(IE,JE)-CON
      B=F(IE,JE+1)-CON
      IF(A*B.GT.0.)GO TO  30
      NIN=1
      EYE=FLOAT(IE)
      JAY=G(JE,A,B)
      GO TO 50
   30 IF(JE.NE.JMAX1)GO TO 39
      IF(IE.NE.IMAX1)GO TO 27
      IF(KONTUR.NE.NCONS.AND.KBEG.LT.80)GO TO 16
C PURGE INITIAL POINTS BUFFER
C SEARCH STORED INITIALPOINTS BUFFER FOR ONR CLOSEST TO PRESENT PEN POSI
      KBEGMX=KBEG-1
      IRET=5
      KBMX02=KBEGMX/2+1
      K=0
   52 K=K+1
      IF(K.LT.KBMX02)GO TO 82
C RETURN IF ALL CONTOURS ARE PLOTTED
      IF(KONTUR.GE.NCONS)RETURN
      KBEG=1
      GO TO 15
   82 IRSQMN=9999999
      DO 55 KBEG=1,KBEGMX
      IF(IBEG(KBEG).EQ.0)GO TO 56
      IRSQ=(I-IBEG(KBEG))**2+(J-JBEG(KBEG))**2
      IF(IRSQ.NE.0)GO TO 53
      IF(CON.NE.CONSV(KBEG))GO TO 53
      IBEG(KBEG)=0
      GO TO 55
   53 IF(IRSQ.GE.IRSQMN)GO TO 56
      IRSQMN=IRSQ
      KBEGMN=KBEG
   56 CONTINUE
C TRACE AND PLOT CLOSEST STARTING CONTOUR
      IF(KBEGMN.LE.0.OR.KBEGMN.GT.100)GO TO 4500
      IE=IBEG(KBEGMN)
      JE=JBEG(KBEGMN)
      CON=CONSV(KBEGMN)
      NIN=NINSV(KBEGMN)
      IBEG(KBEGMN)=0
      DO 57 MP=1,IDFS
   57 FS(MP)=0
      IF(NIN.LT.1.OR.NIN.GT.4)GO TO 4500
      GO TO (40,41,42,43),NIN
   58 IF(J.EQ.0)J=1
      IF(I.EQ.0)I=1
      IF(I.EQ.IMAX)I=IMAX1
      IF(J.EQ.JMAX)J=JMAX1
      GO TO 52
```

```
CURVE FOLLOW BEGIN ROUTINE
C IF THERE IS ROOM , SAVE THE INITIAL POINT
   50 IF(KBEG.GT.100)LPLOT=.TRUE.
      IF(LPLOT)GO TO 59
      CONSV(KBEG)=CON
      IBEG(KBEG)=IE
      JBEG(KBEG)=JE
      NINSV(KBEG)=NIN
      KBEG=KBEG+1
C     IF(KBEG.GT.100)GO TO 450)
   59 IBIT=IE+IMAX*(JE-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      IF( AND(MASK(IBIT),FS(IWORD)).NE.0)GO TO (22,24,26,30),IRET
      KLING = .FALSE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING =.TRUE.
      I=IE
      J=JE
C
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(NONO.LT.0)GO TO 1000
      IF(KLING) GO TO 1100
      IF (LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NOGT)
C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C    1A,NOGT)
 1000 CONTINUE
C     IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,3)
      IF(LPLOT)CALL PLOT(EKEY,EKAY,3)
C
1100  CONTINUE
      KLING = .TRUE.
      EYEZRO=EYE
      JAYZRO=JAY
      KLOG=.FALSE.
      IPEN=2
      HALT=.FALSE.
CURVE FALLOWING ROUTINE
   60 A=F(I,J)-CON
      B=F(I,J+1)-CON
      C=F(I+1,J+1)-CON
      D=F(I+1,J)-CON
      IBIT=I+IMAX*(J-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      FS(IWORD)= OR(FS(IWORD),MASK(IBIT))
      JPP=1
      IF(A*C.GE.0.)JPP=JPP+1
      IF(B*D.GE.0.)JPP=JPP+2
      IF(JPP.EQ.4)GO TO 44
   73 NOUT=NPT(NIN,JPP)
      GO TO(72,74,76,78),NOUT
   44 IF(A*B)45,46,47
   46 JPP=2
      GO TO 73
C  47 IM1=I-1
C     JM1=J-1
C     IP2=I+2
C     JP2=J+2
C    WRITE(6,45) I,J,NIN,CON,((F(IZ,JZ),IZ=IM1,IP2),JZ=JM1,JP2)
C  45 FORMAT(23H1ERROR IN CONTOR FOR I=,I2(3H)J=,I2,5H,NIN=,I1,12H,AND
C    1CONTOR=,E18.8//(4E20.8))
   47 GO TO (22,24,26,30,58),IRET
   72 EYE=FLOAT(I)
      JAY=G(J,A,B)
      IF(I.EQ.1)HALT=.TRUE.
      I=I-1
      GO TO 80
   74 EYE=G(I,B,C)
      JAY=FLOAT(J+1)
      IF(J.EQ.JMAX1)HALT=.TRUE.
      J=J+1
      GO TO 80
```

```
   76 EYE=FLOAT(I+1)
      JAY=G(J,D,C)
      IF(I.EQ.IMAX1)HALT=.TRUE.
      I=I+1
      GO TO 80
   78 EYE=G(I,A,D)
      JAY=FLOAT(J)
      IF(J.EQ.1)HALT=.TRUE.
      J=J-1
C  80 IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,IPEN)
C
   80 CONTINUE
      KLING = .TRUE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING=.FALSE.
       THING = .FALSE.
       IF(IPEN.EQ.3) THING = .TRUE.
       IPEN = 3
       IF(KLING) IPEN = 2
       IF(THING) IPEN = 3
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(LPLOT)CALL PLOT(EKEY,EKAY,IPEN)
      IF(KLOG.AND.EYE.EQ.EYEZRO.AND.JAY.EQ.JAYZRO)HALT=.TRUE.
      IF(HALT)GO TO 90
      IPEN=1
      NIN=NPT(NOUT,1)
      KLOG=.TRUE.
      GO TO 60
CURVE FOLLOW END
   90 IF(NONO.LT.0)GO TO 1001
      IF(IPEN.EQ.3) GO TO 1001
      IF(LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C    1A,NDGT)
 1001 CONTINUE
      GO TO(22,24,26,30,58),IRET
 4500 WRITE(6,4501)NIN,KBEG,KBEGMX,K,IE,JE,KONTUR,CON,KBEGMN
 4501 FORMAT(4HNIN=,I15/5HKBEG=,I15/7HKBEGMX=,I15/2HK=,I15/3HIE=,I15/
     *3HJE=,I15/7HKONTUR=I15/4HCON=,E20.8/ 7HKBEGMN=,I15//)
      KBEG=1
      GO TO 15
7000  WRITE(6,7001)
7001  FORMAT(1H1/5X,"ERROR-NEGATIVE NO. OF CONTOURS"/1H1)
      RETURN
      END
```

D. <u>E/EPM CONTOUR PLOTTING PROGRAM</u>

This program produces a contour plot of E/EPM. The boundaries are distance in the X and Y direction. The region corresponds to the grid the wave model uses.The box that is drawn around the contour should reflect the boundaries of the section of the ocean that the model is running in.

The program starts out by reading in preliminary information about grid size, CONTUN values, etc. READS is then called to load the buffer with grid point data. Next, a pair of arrays that CONTUN needs are zeroed out. F is then loaded with the correct values. Just prior to calling CONTUN the region is boxed in and the title is placed on it. After CONTUN the last plot buffer is dumped and then the data that was read in is written out for verification purposes.

By changing primarily one line of code this program can be altered so it displays SIGNIFICANT WAVE HEIGHT. After the E/EPM code and plot is a altered listing and example plot.

INPUT DATA LIST FOR E/EPM CONTOUR PLOTTING PROGRAM

FORMAT	VARIABLE	DESCRIPTION
I10,7A10	NCHAR,TITLE	number of characters in title,title of plot
I10	NB	number of words to be read in at a time
I10	NF	number of frequencies
2I10	IGRDX,IGRDY	number of grid points in x direction, in y direction
2I10	NPTSK,IOFSET	number of grid point values to skip, number of values prior to first grid point value
I10,3F10.5	MODE,AR	mode flag, limit values
2F10.5	SI,SJ	number of points per inch; in x direction, y direction
I10,F10.5	NONO,THETA	label flag, if NONO=0 then no labels else labels are present on each contour; angle of rotation in degrees of label
I10,F10.5	NDGT,SIZE	number of decimal places to appear in labels, size in inches of label(vertical height of char)

VARIABLE LIST FOR E/EPM CONTOUR PLOTTING PROGRAM

EPM/60522.0/ - value used to normalize energy
NCHAR - number of characters in title
TITLE - title of plot
NB - number of words to be read in, size of buffer
NF - number of frequencies
IGRDX - number of grid points in x direction
IGRDY - number of grid points in Y direction
NPTSK - number of grid point values to skip
IOFSET - number of words of information prior to the first grid point value
MODE - mode flag for CONTUN (see CONTUN for more info)
AR - limits for CONTUN
SI - number of points to plotted per inch in X direction
SJ - number of points to be plotted per inch in the Y direction
NONO - label flag; if NONO=0 then contours are not labeled else contours are labeled,
THETA - angle of rotation in degrees of label
NDGT - number of digits to left of decimal point to be put on each label
SIZE - size in inches of label (vertical height of numbers)
BUFFER - input buffer, used to hold SREC values before they are put in F
FS - mask array for CONTUN, must be zeroed out
FSMSK - mask array for CONTUN, must be zeroed out
IND - used as index into BUFFER
J - do loop index
I - do loop index
F - array used by CONTUN for plotting information, loaded with altered values from BUFFER
XMAX - length of X axis in inches
YMAX - length of Y axis in inches
XINDT - x coordinate of new origin
YINDT - Y coordinate of new origin

READS- this subroutine reads an SREC

PARAMETERS-

LUN - logical unit number, device number of tape with SREC
BUFFER - input buffer that is to receive the SREC data
NB - number of words to be read in
NF - number of frequencies

LOCAL VARIABLES-

I - do loop index
DUM - temp location used to read into while skipping over records

TEXT-

This subroutine spaces the tape forward to the SREC and then loads it into BUFFER. The number of words read in depends on NB,

PROGRAM EPM

```
      PROGRAM EPM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=1,TAPE7)
C
C AR - INTERVAL VALUES FOR /CONTUN/
C F - INPUT ARRAY FOR /CONTUN/
C FREQ - ARRAY OF FREQUENCIES
C FS - MASK ARRAY
C FSMSK - MASK ARRAY FOR CONTUN
C IGRDX - NUMBER OF GRID POINTS IN X DIRECTION
C IGRDY - NUMBER OF GRID POINTS IN Y DIRECTION
C MODE - SPECIFIES MODE FOR /CONTUN/
C NDGT - NUMBER OF DIGITS IN CONTOUR LABEL
C NF - NUMBER OF FREQUENCIES
C NONO - FLAG TO TELL /CONTUN/ WHETHER OR NOT TO LABEL CONTOURS
C NPTSK - NUMBER OF GRID POINTS TO SKIP
C SI - NUMBER OF POINTS PER INCH IN X DIRECTION
C SJ - NUMBER OF POINTS PER INCH IN Y DIRECTION
C SIZE - HEIGHT OF CONTOUR LABELS AS PART OF AN INCH
C THETA - ANGLE THAT CONTOUR LABELS ARE ROTATED
      REAL F(20,20),BUFFER(1511)
      REAL AR(3),SI,SJ,THETA,SIZE
      REAL FSMSK(530),XMAX,YMAX,XINDT,YINDT
      INTEGER NB,NF,IGRDX,IGRDY,IOFSET,NPTSK,MODE,FS(18),TITLE(7)
      DATA EPM/60522.0/
C READ IN LENGTH OF TITLE AND TITLE OF GRAPH
      READ(5,1000) NCHAR,(TITLE(I),I=1,7,1)
C     GET SIZE OF BUFFER
      READ(5,1010) NB
C     GET NUMBER OF FRUENCIES
      READ(5,1010) NF
C     GET X AND Y DIMENSIONS OF GRID
      READ(5,1010) IGRDX,IGRDY
C     GET NUMBER OF GRID POINTS TO SKIP AND IOFSET
      READ(5,1010) NPTSK,IOFSET
C     GET MODE AND AR VALUES
      READ(5,1020) MODE,AR(1),AR(2),AR(3)
C     POINTS PER INCH
      READ(5,1030) SI,SJ
C GET LABELING FLAG AND ANGLE OF ROTATION FOR LABELS
      READ(5,1020) NONO,THETA
C     GET NUMBER OF DIGITS AND SIZE OF LABELS
      READ(5,1020) NDGT,SIZE
 1000 FORMAT(I10,7A10)
 1010 FORMAT(8I10)
 1020 FORMAT(I10,7F10.0)
 1030 FORMAT(8F10.0)
C     LOAD BUFFER
      CALL READS(1,BUFFER,NB,NF)
      DO 98 I=1,18,1
         FS(I)=0
         FSMSK(I)=0.0
   98    CONTINUE
      DO 99 I=19,530,1
         FSMSK(I)=0.0
   99 CONTINUE
C     ADJUST BUFFER INDEX
      IND = IOFSET + NPTSK + 1
C     LOAD F
      DO 10 J=1,IGRDY,1
         DO 10 I=1,IGRDX,1
      F(I,J)=BUFFER(IND)/EPM
            IND = IND + 1
   10    CONTINUE
C     GET DIMENSIONS OF PLOT
      XMAX=IGRDX/SI
      YMAX=IGRDY/SJ
C     INITIALIZE THE PLOTTER
      CALL PLOTS(0,0,7)
C     MOVE PEN AWAY FROM EDGE OF PAPER
      CALL PLOT(0.5,0.5,-3)
C     DRAW BOX AROUND PLOT
      CALL PLOT(0.0,YMAX,2)
      CALL PLOT(XMAX,YMAX,2)
      CALL PLOT(XMAX,0.0,2)
      CALL PLOT(0.0,0.0,2)
C     SET UP NEW ORIGIN
```

```
      XINDT=1.0/(2.0*SI)
      YINDT=1.0/(2.0*SJ)
      CALL PLOT(XINDT,YINDT,-3)
C     PUT TITLE ON GRAPH
      CALL SYMBOL(0.0,YMAX,0.25,TITLE,0.0,NCHAR)
C     DO CONTOUR PLOT           FINALLY]]]]]]
      CALL CONTUN(F,IGRDX,IGRDY,FS,AR,MODE,SI,SJ,THETA,NONO,SIZE,
     +            FSMSK,NDGT)
      CALL PLOT(0,0,999)
C     DUMP DATA
      WRITE(6,220) THETA,NDGT,SIZE
  220 FORMAT(7H THETA=,F10.6,5X,5HNDGT=,I4,5X,5HSIZE=,F10.6)
      WRITE(6,230) XMAX,YMAX
  230 FORMAT(13H SIZE OF BOX:,F10.6,3H X ,F10.6)
      WRITE(6,300) NB
  300 FORMAT(4H NB=,I4)
      WRITE(6,301) NF
  301 FORMAT(4H NF=,I4)
      WRITE(6,302) IGRDX,IGRDY
  302 FORMAT(9H GRID IS ,I4,3H X ,I4)
      STOP
      END
```

```
      SUBROUTINE READS (LUN, BUFFER, NB, NF)
C
C     SUBROUTINE TO READ SUMMARY RECORD - -
C          SUMS OVER FREQUENCY AND DIRECTION.
C
      REAL BUFFER(NB)
C
      REWIND LUN
C     SKIP TO SREC.
      DO 10 I=1,NF
      BUFFER IN (LUN,1) (DUM,DUM)
C
      IF (UNIT (LUN)) 10, 50, 50
C
 10   CONTINUE
C     READ SREC.
      BUFFER IN (LUN,1) (BUFFER(1),BUFFER(NB))
C
      IF (UNIT (LUN)) 20, 50, 50
C
 20   CONTINUE
      REWIND LUN
C
      RETURN
C
 50   WRITE (6,1000) LUN
 1000 FORMAT (32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
C
      STOP
C
      END
```

```
      SUBROUTINE CONTUN(F,IMAX,JMAX,FS,AR,MODE,SCALEI,SCALEJ,THETA,NONO,
     1SIZE,FSMSK,NDGT)
C
C       TRACES CONTOURS THROUGH ARRAY F(IMAX,JMAX) AND PLOTS THEM
C                 WHITH SCALEI,SCALEJ POINTS PR. CM. THE CONTOURS ARE
C                 LABELED AT AN ANGLE THETA WITH THE X-AXES PROVIDED
C                 NONO =AN INTEGER .GE.0. IF NONO.LT.0 THE CONTOURS ARE
C                 NOT LABELED. SIZE IS THE SIZE OF THE LABEL.
C
C        MODE = 1 FAMILY OF CONTOURS IS TRACED WITH INTERVAL=AR(1).
C        MODE = 2 FAMILY OF APPROXIMATELY AR(1)  CONTOURS IS TRACED WITH
C                 A REASONABLY COMPUTED CONTOUR INTERVAL.
C        MODE = 3 A SINGLE CONTOUR WITH VALUE AR(1) IS TRACED.
C        MODE = 4 AR(1)=INTERVAL,AR(2)=MIN,AR(3)=MAX
C        MODE = 5 AR(1)=NR OF CONTOURS, AR(2)*FMAX=MIN,AR(3)*FMAX=MAX
C
C        IMAX COINCIDES WITH THE PAPER X-AXES. DIMENSION OF ARRAY IN X DIR
C        JMAX COINCIDES WITH THE PAPER Y-AXES. DIMENSION OF ARRAY IN Y DIR
C
C        FS  AN INTEGER ARRAYS MUST BE GIVEN DIMENSIONS 1+IMAX*JMAX/31 IN
C                 CALLING PROGRAM.
C
C        NDGT=SIGNIFICANCE,IF NEG. IS COMPUTED.
C 2 S COMPLEMENT ARITMETIC
C EXTERNAL FUNTION LAND I,J$ IS THE BOOLEAN  I.E.LOGICAL$ AND OF TWO
C                 FULLWORD INTEGERS.
C EXTERNAL FUNCTION LOR I,J$ IS THE BOOLEAN OR OF TWO FULLWORD INTEGERS.
C
C SPECIFICATION STATEMENTS.
C
C      CHANGE KDS 5.1.73
      INTEGER        FS(1)
      REAL JAY,JAYZRO,F(IMAX,JMAX)
      DIMENSION MASK(31),NPT(4,3),AR(3),AC(5),ALG(5)
      DIMENSION IBEG(100),JBEG(100),CONSV(100),NINSV(100)
      INTEGER FSMSK(1)
      LOGICAL HALT,LPLOT,KLOG
      LOGICAL KLING,THING
      DATA NPT/3,4,1,2,2,1,4,3,4,3,2,1/,IFLAG/0/
      DATA AC/1.25,2.,2.5,5.,10./
      DATA ALG/.097,.3,.398,.7,1.0/
C FUNCTION DEFINITION.
      G(I,A,B)=FLOAT(I)-A/(B-A)
C SET PLOTTING PARAMETERS.
      IDFS=JMAX*IMAX
      IDFS=IDFS/31+1
      ARG2=AR(1)
      KBEG=1
      KBEGMX=0
      CSALEI=1./SCALEI
      CSALEJ=1./SCALEJ
C     IF (IOUT.NE.1) GO TO 604
C     DO 602 I=1,IMAX
C     WRITE (6,603) (F(I,J),J=1,JMAX)
C 602 CONTINUE
C 603 FORMAT(1H ,21F6.0)
C 604 CONTINUE
      IF (IFLAG.EQ.5) GO TO 4
      DO 3 I=1,31
    3 MASK(I)=2**(31-I)
C
C SET UP LOOP CONTROLLING SELECTION OF CONTOURS
C
      IFLAG=5
    4 IMAX1=IMAX-1
      JMAX1=JMAX-1
      IF (MODE.NE.3) GO TO 10
      FMIN=ARG2
      CONINT=ARG2
      NCONS=1
      GO TO 15
   10 FMAX=F(1,1)
      FMIN=FMAX
      DO 12 I=1,IMAX
      DO 12 J=1,JMAX
```

```
      FMIN=AMIN1(FMIN,F(I,J))
   12 FMAX=AMAX1(FMAX,F(I,J))
      IF (MODE.NE.5) GO TO 120
      AR(3)=AR(3)*FMAX
      AR(2)=AR(2)*FMAX
      ARG2=(AR(3)-AR(2))/AR(1)
  120 CONINT=ARG2
      IF (MODE.NE.2) GO TO 13
C IF MODE =2 SELECT A CONTOUR INTERVAL
      ALGCNT=ALOG10((FMAX-FMIN)/ARG2)
      N=ALGCNT
      ALGCNT=AMOD(ALGCNT,1 )
      IF (ALGCNT.GE.0.) GO TO 18
      N=N-1
      ALGCNT=1.+ALGCNT
   18 B=ALGCNT
      A=1.
      DO 19 I=1,5
      C=ABS(ALGCNT-ALG(I))
      IF (C.GE.B) GO TO 19
      B=C
      A=AC(I)
   19 CONTINUE
      CONINT=A*10.**N
C DETERMINE NUMBER OF CONTOURS AND MINIMUM VALUE
   13 IF (FMIN.GT.0.) FMIN=FMIN+CONINT
      FMIN=CONINT*AINT(FMIN/CONINT)
      IF (FMAX.LT.0.) FMAX=FMAX-CONINT
      FMAX=CONINT*AINT(FMAX/CONINT)
      IF (MODE.LT.4) GO TO 11
      FMIN=AMAX1(FMIN,AR(2))
      FMAX=AMIN1(FMAX,AR(3))
   11 NCONS=(FMAX-FMIN)/CONINT+1.
      NCONS=MINO(NCONS,40)
      IF(NCONS.LE.0) GO TO 7000
C DETERMINE NUMBER OF DIGITS IN LABEL
   15 DEL=1.E-4*CONINT
      IF(NDGT.GE.0) GO TO 20
      NDGT=0
   21 IF (ABS(AMOD(CONINT*10.**NDGT,1.)).LT.DEL) GO TO 20
      NDGT=NDGT+1
      IF (NDGT.LT.11) GO TO 21
C INSURE THAT NO POINT IS EXACTLY ON A CONTOUR
   20 DO 32 I=1,IMAX
      DO 32 J=1,JMAX
   32 IF (ABS(AMOD(F(I,J),CONINT)).LT.DEL) F(I,J)=1.000001*F(I,J)+2.*DE
C START CONTOUR PLOTTING
      KONTUR=0
   16 KONTUR=KONTUR+1
      CON=FMIN+CONINT*FLOAT(KONTUR-1)
C     WRITE (6,620) CON,FMIN,CONINT,KONTUR
C 620 FORMAT(1H ,5X,  CON   ,F10.4,  FMIN   ,F10.4,  CONINT   ,F10.4,  KONT
C    1UR   ,I8)
      DO 14 MP=1,IDFS
   14 FS(MP)=0
C
C BEGIN EDGE SEARCH
C
      LPLOT=.FALSE.
      JE=0
   17 JE=JE+1
      IRET=1
   42 D=F(IMAX,JE)-CON
      C=F(IMAX,JE+1)-CON
      IF (C*D.GT.0.) GO TO 22
      IE=IMAX1
      NIN=3
      EYE=FLOAT(IMAX)
      JAY=G(JE,D,C)
      GO TO 50
   22 IF (JE.NE.JMAX1) GO TO 17
      IE=0
   23 IE=IE+1
      IRET=2
   43 A=F(IE,1)-CON
```

```
      D=F(IE+1,1)-CON
      IF (A*D.GT.0.) GO TO 24
      NIN=4
      JE=1
      EYE=G(IE,A,D)
      JAY=1.
      GO TO 50
   24 IRET=3
   41 B=F(IE,JMAX)-CON
      C=F(IE+1,JMAX)-CON
      IF(B*C.GT.0.)GO TO 26
      NIN=2
      JE=JMAX1
      EYE=G(IE,B,C)
      JAY =FLOAT(JMAX)
      GO TO 50
   26 IF(IE.NE.IMAX1)GO TO 23
C  BEGIN INTERIOR SEARCH
      IE=0
   27 IE=IE+1
      LPLOT=.TRUE.
      IF(IE.EQ.1)LPLOT=.FALSE.
      JE=0
   39 JE=JE+1
      IRET=4
   40 A=F(IE,JE)-CON
      B=F(IE,JE+1)-CON
      IF(A*B.GT.0.)GO TO  30
      NIN=1
      EYE=FLOAT(IE)
      JAY=G(JE,A,B)
      GO TO 50
   30 IF(JE.NE.JMAX1)GO TO 39
      IF(IE.NE.IMAX1)GO TO 27
      IF(KONTUR.NE.NCONS.AND.KBEG.LT.80)GO TO 16
C PURGE INITIAL POINTS BUFFER
C SEARCH STORED INITIALPOINTS BUFFER FOR ONR CLOSEST TO PRESENT PEN POSI
      KBEGMX=KBEG-1
      IRET=5
      KBMXO2=KBEGMX/2+1
      K=0
   52 K=K+1
      IF(K.LT.KBMXO2)GO TO 82
C RETURN IF ALL CONTOURS ARE PLOTTED
      IF(KONTUR.GE.NCONS)RETURN
      KBEG=1
      GO TO 15
   82 IRSQMN=9999999
      DO 56 KBEG=1,KBEGMX
      IF(IBEG(KBEG).EQ.0)GO TO 56
      IRSQ=(I-IBEG(KBEG))**2+(J-JBEG(KBEG))**2
      IF(IRSQ.NE.0)GO TO 53
      IF(CON.NE.CONSV(KBEG))GO TO 53
      IBEG(KBEG)=0
      GO TO 55
   53 IF(IRSQ.GE.IRSQMN)GO TO 56
      IRSQMN=IRSQ
      KBEGMN=KBEG
   56 CONTINUE
C TRACE AND PLOT CLOSEST STARTING CONTOUR
      IF(KBEGMN.LE.0.OR.KBEGMN.GT.100)GO TO 4500
      IE=IBEG(KBEGMN)
      JE=JBEG(KBEGMN)
      CON=CONSV(KBEGMN)
      NIN=NINSV(KBEGMN)
      IBEG(KBEGMN)=0
      DO 57 MP=1,IDFS
   57 FS(MP)=0
      IF(NIN.LT.1.OR.NIN.GT.4)GO TO 4500
      GO TO (40,41,42,43),NIN
   58 IF(J.EQ.0)J=1
      IF(I.EQ.0)I=1
      IF(I.EQ.IMAX)I=IMAX1
      IF(J.EQ.JMAX)J=JMAX1
      GO TO 52
```

```
CURVE FOLLOW BEGIN ROUTINE
C IF THERE IS ROOM , SAVE THE INITIAL POINT
   50 IF(KBEG.GT.100)LPLOT=.TRUE.
      IF(LPLOT)GO TO 59
      CONSV(KBEG)=CON
      IBEG(KBEG)=IE
      JBEG(KBEG)=JE
      NINSV(KBEG)=NIN
      KBEG=KBEG+1
C     IF(KBEG.GT.100)GO TO 450)
   59 IBIT=IE+IMAX*(JE-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      IF( AND(MASK(IBIT),FS(IWORD)).NE.0)GO TO (22,24,26,30),IRET
      KLING = .FALSE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING =.TRUE.
      I=IE
      J=JE
C
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(NOND.LT.0)GO TO 1000
      IF(KLING) GO TO 1100
      IF (LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C    1A,NDGT)
 1000 CONTINUE
C     IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,3)
      IF(LPLOT)CALL PLOT(EKEY,EKAY,3)
C
1100  CONTINUE
      KLING = .TRUE.
      EYEZRO=EYE
      JAYZRO=JAY
      KLOG=.FALSE.
      IPEN=2
      HALT=.FALSE.
CURVE FALLOWING ROUTINE
   60 A=F(I,J)-CON
      B=F(I,J+1)-CON
      C=F(I+1,J+1)-CON
      D=F(I+1,J)-CON
      IBIT=I+IMAX*(J-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      FS(IWORD)= OR(FS(IWORD),MASK(IBIT))
      JPP=1
      IF(A*C.GE.0.)JPP=JPP+1
      IF(B*D.GE.0.)JPP=JPP+2
      IF(JPP.EQ.4)GO TO 44
   73 NOUT=NPT(NIN,JPP)
      GO TO(72,74,76,78),NOUT
   44 IF(A*B)45,46,47
   46 JPP=2
      GO TO 73
C  47 IM1=I-1
C     JM1=J-1
C     IP2=I+2
C     JP2=J+2
C     WRITE(6,45) I,J,NIN,CON,((F(IZ,JZ),IZ=IM1,IP2),JZ=JM1,JP2)
C  45 FORMAT(23H1ERROR IN CONTUR FOR I=,I2(3H J=,I2,5H,NIN=,I1,12H,AND
C    1CONTUR=,E18.8//(4E20.8))
   47 GO TO (22,24,26,30,58),IRET
   72 EYE=FLOAT(I)
      JAY=G(J,A,B)
      IF(I.EQ.1)HALT=.TRUE.
      I=I-1
      GO TO 80
   74 EYE=G(I,B,C)
      JAY=FLOAT(J+1)
      IF(J.EQ.JMAX1)HALT=.TRUE.
      J=J+1
      GO TO 80
```

```
   76 EYE=FLOAT(I+1)
      JAY=G(J,3,C)
      IF(I.EQ.IMAX1)HALT=.TRUE.
      I=I+1
      GO TO 80
   78 EYE=G(I,4,D)
      JAY=FLOAT(J)
      IF(J.EQ.1)HALT=.TRUE.
      J=J-1
C  80 IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,IPEN)
C
   80 CONTINUE
      KLING = .TRUE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING=.FALSE.
       THING = .FALSE.
       IF(IPEN.EQ.3) THING = .TRUE.
       IPEN = 3
       IF(KLING) IPEN = 2
       IF(THING) IPEN = 3
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(LPLOT)CALL PLOT(EKEY,EKAY,IPEN)
      IF(KLOG.AND.EYE.EQ.EYEZRO.AND.JAY.EQ.JAYZRO)HALT=.TRUE.
      IF(HALT)GO TO 90
      IPEN=1
      NIN=NPT(NOUT,1)
      KLOG=.TRUE.
      GO TO 60
CURVE FOLLOW END
   90 IF(NDND.LT.0)GO TO 1001
      IF(IPEN.EQ.3) GO TO 1001
      IF(LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THE
C    1A,NDGT)
 1001 CONTINUE
      GO TO(22,24,26,30,58),IRET
 4500 WRITE(6,4501)NIN,KBEG,KBEGMX,K,IE,JE,KONTUR,CON,KBEGMN
 4501 FORMAT(4HNIN=,I15/5HKBEG=,I15/7HKBEGMX=,I15/2HK=,I15/3HIE=,I15/
     *3HJE=,I15/7HKONTUR=I15/4HCON=,E20.8/ 7HKBEGMN=,I15//)
      KBEG=1
      GO TO 15
 7000 WRITE(6,7001)
 7001 FORMAT(1H1/5X,"ERROR-NEGATIVE NO. OF CONTOURS"/1H1)
      RETURN
      END
```

PROGRAM SW

```
      PROGRAM SW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=1,TAPE7)
C
C AR - INTERVAL VALUES FOR /CONTUN/
C F - INPUT ARRAY FOR /CONTUN/
C FREQ - ARRAY OF FREQUENCIES
C FS - MASK ARRAY
C FSMSK - MASK ARRAY FOR CONTUN
C IGRDX - NUMBER OF GRID POINTS IN X DIRECTION
C IGRDY - NUMBER OF GRID POINTS IN Y DIRECTION
C MODE - SPECIFIES MODE FOR /CONTUN/
C NDGT - NUMBER OF DIGITS IN CONTOUR LABEL
C NF - NUMBER OF FREQUENCIES
C NONO - FLAG TO TELL /CONTUN/ WHETHER OR NOT TO LABEL CONTOURS
C NPTSK - NUMBER OF GRID POINTS TO SKIP
C SI - NUMBER OF POINTS PER INCH IN X DIRECTION
C SJ - NUMBER OF POINTS PER INCH IN Y DIRECTION
C SIZE - HEIGHT OF CONTOUR LABELS AS PART OF AN INCH
C THETA - ANGLE THAT CONTOUR LABELS ARE ROTATED
      REAL BUFFER(1511)
      REAL F(19,25)
      REAL AR(3),SI,SJ,THETA,SIZE
      REAL XMAX,YMAX,XINDT,YINDT
      INTEGER FSMSK(18)
      INTEGER NB,NF,IGRDX,IGRDY,IOFSET,NPTSK,MODE,FS(18),TITLE(7)
C READ IN LENGTH OF TITLE AND TITLE OF GRAPH
      READ(5,1000) NCHAR,(TITLE(I),I=1,7,1)
      WRITE(6,306) TITLE
  306 FORMAT(7H TITLE:,/,1H ,10A10)
C     GET SIZE OF BUFFER
      READ(5,1010) NB
      WRITE(6,300) NB
  300 FORMAT(4H NB=,I4)
C     GET NUMBER OF FRUENCIES
      READ(5,1010) NF
      WRITE(6,301) NF
  301 FORMAT(4H NF=,I4)
C     GET X AND Y DIMENSIONS OF GRID
      READ(5,1010) IGRDX,IGRDY
      WRITE(6,302) IGRDX,IGRDY
  302 FORMAT(9H GRID IS ,I4,3H X ,I4)
C     GET NUMBER OF GRID POINTS TO SKIP AND IOFSET
      READ(5,1010) NPTSK,IOFSET
C     GET MODE AND AR VALUES
      READ(5,1020) MODE,AR(1),AR(2),AR(3)
      WRITE(6,305) MODE,AR
  305 FORMAT(6H MODE=,I10,5X,3HAR=,3(F10.5,5X))
C     POINTS PER INCH
      READ(5,1030) SI,SJ
C GET LABELING FLAG AND ANGLE OF ROTATION FOR LABELS
      READ(5,1020) NONO,THETA
C     GET NUMBER OF DIGITS AND SIZE OF LABELS
      READ(5,1020) NDGT,SIZE
      WRITE(6,220) THETA,NDGT,SIZE
  220 FORMAT(7H THETA=,F10.6,5X,5HNDGT=,I4,5X,5HSIZE=,F10.6)
      READ(5,1030) XMAX,YMAX
      WRITE(6,230) XMAX,YMAX
  230 FORMAT(13H SIZE OF BOX:,F10.6,3H X ,F10.6)
      READ(5,1030) XINDT,YINDT
      WRITE(6,304) XINDT,YINDT
  304 FORMAT(7H XINDT=,F10.5,10X,6HYINDT,F10.5)
 1000 FORMAT(I10,7A10)
 1010 FORMAT(8I10)
 1020 FORMAT(I10,7F10.0)
 1030 FORMAT(8F10.0)
C     LOAD BUFFER
      CALL READS(1,BUFFER,NB,NF)
      DO 98 I=1,18,1
         FS(I)=0
         FSMSK(I)=0
   98    CONTINUE
C     ADJUST BUFFER INDEK
      IND = IOFSET + NPTSK + 1
C     LOAD F
      DO 10 J=1,IGRDY,1
         DO 10 I=1,IGRDX,1
```

PROGRAM SW

```
                   F(I,J)=0.04*SQRT(BUFFER(IND))
                   IND = IND + 1
   10           CONTINUE
C          SET DIMENSIONS OF PLOT
C          INITIALIZE THE PLOTTER
           CALL PLOTS(0,0,7)
C          MOVE PEN AWAY FROM EDGE OF PAPER
           CALL PLOT(0.5,0.5,-3)
C          DRAW BOX AROUND PLOT
           CALL PLOT(0.0,YMAX,2)
           CALL PLOT(XMAX,YMAX,2)
           CALL PLOT(XMAX,0.0,2)
           CALL PLOT(0.0,0.0,2)
C          SET UP NEW ORIGIN
           CALL PLOT(XINDT,YINDT,-3)
C          PUT TITLE ON GRAPH
           CALL SYMBOL(0.0,YMAX,0.25,TITLE,0.0,NCHAR)
C          DO CONTOUR PLOT              FINALLY]]]]]]
           CALL CONTUN(F,IGRDX,IGRDY,FS,AR,MODE,SI,SJ,THETA,NONO,SIZE,
          +       FSMSK,NDGT)
           CALL PLOT(0,0,999)
           STOP
           END
```

E. PEAK FREQUENCY CONTOUR PLOTTING PROGRAM

This program produces a contour plot of peak frequency over a rectangular region. The boundaries as marked by a box that this program draws around the contour are lengths. The box is supposed to be the gridded area produced by the wave model. As of now the program will handle a 20 x 20 grid.

The program begins by reading in some preliminary information. READJS is then called to load S1D with the raw data values. Next the program draws the box and titles it. The next section of the program is a routine to determine the peak frequency. If the peak is not an end point then the point nearest the peak and the two around it are used to determine a new peak frequency. After this the program zeroes out some mask arrays for CONTUN and then it calls CONTUN. Just prior to calling CONTUN the input is written out for verification purposes.

INPUT DATA LIST FOR PEAK FREQUENCY CONTOUR PLOTTING PROGRAM

FORMAT	VARIABLE	DESCRIPTION
I10	NDU	non-dimensional unit flag; if NDU=1 then non-dimensional units will be used, else dimensional units will be used
I10,7A10	NCHAR,TITLE	number of characters in title, title of plot
I10,7F10.5,/,(8F10.5)		NF,FREQ number of frequencies,followed by NF frequency values
I10	NPTSK	number of grid points to skip
2I10	IGRDX,IGRDY	number of grid points, in x direction, Y direction
I10,3F10.5	MODE,AR	mode flag and limit values for CONTUN
I10,F10.5	NONO,THETA	label flag, if NONO=0 then contours are not labeled else they are labeled; angle of rotation for label
I10,F10.5	NDGT,SIZE	number of decimal places to be used for labels; height of digits in label in inches
2F10.5	SI,SJ	number of points to be plotted per inch, x direction, y direction

LIST OF VARIABLES USED IN PEAK FREQUENCY CONTOUR PLOTTING PROGRAM

NDU - non-dimensional units flag, if NDU=1 then non-dimensional units are used else dimensional units are retained
LFPM - number used to normalize frequency
NCHAR - number of characters in title
TITLE - title of plot
NF - number of frequencies
FREQ - list of actual frequency values
NPTSK - number of grid point values to skip
IGRDX - number of grid points in x direction
IGRDY - number of grid points in Y direction
MODE - mode flag for CONTUN (see CONTUN for more info)
AR - limit values for CONTUN (see CONTUN for more info)
NONO - labeling flag, if NONO=0 then contours are not labeled, if NONO≠0 then contours are labeled
THETA - angle of rotation of contour labels
NDGT - number of digits to left of decimal point to be displayed on contour labels
SIZE - vertical height of digits of contour labels
SI - number of points plotted per inch in X direction
SJ - number of points plotted per inch in Y direction
NGRID - number of points to be transferred into S1D
S1D - array 1-D spectrum values
IND - index into S1D
XMAX - length of x axis in inches
YMAX - length of Y axis in inches
XINDT - x coordinate of new origin
YINDT - y coordinate of new origin
L - do loop index
K - do loop index
SM - used to hold the S1D value that seems to be the largest
JM - the index of the largest value
X1 - frequency at point prior to peak
Y1 - S1D value at point prior to peak
X2 - frequency at peak
Y2 - S1D value at peak
X3 - frequency at point just past peak
Y3 - S1D value just after peak
DENOM - denominator of expression used to calculate peak frequency
F - array used by CONTUN to do contour plot, filled with frequency values at each grid point
XNUM - numerator of curve fit equation
FS - mask array used by CONTUN, must be zeroed out
FSMSK - mask array used by CONTUN, must be zeroed out

READJS- this subroutine reads a JSREC

PARAMETERS-

LUN - logical unit number of file containg JSREC
S1D - array to be loaded with 1-D spectra values
NF - number of frequencies
NGRID - number of values to be loaded into S1D

LOCAL VARIABLES-

J - do loop index
BUFFER - input buffer
I - do loop index

TEXT-

This subroutine loads the 1-D spectra values into S1D and leaves the tape containing the JSREC rewound to beginning of file.

```
      PROGRAM PKFRCON(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     +               TAPE1=1)
C
C   THIS PROGRAM DOES CONTOUR PLOTS OF PEAK FREQUENCY OVER THE WHOLE GRID
C
C TITLE - TITLE OF GRAPH
C NCHAR - NUMBER OF CHARACTERS IN TITLE
C NF - NUMBER OF FREQUENCIES
C IGRDX - NUMBER OF GRID POINTS IN X DIRECTION
C IGRDY - NUMBER OF GRID POINTS IN Y DIRECTION
C NPTSK - NUMBER OF GRID POINTS TO SKIP
C MODE - SPECIFIES MODE FOR /CONTUN/
C AR - INTERVAL VALUES FOR /CONTUN/
C SI - NUMBER OF POINTS PER INCH IN X DIRECTION
C SJ - NUMBER OF POINTS PER INCH IN Y DIRECTION
C THETA - ANGLE THAT CONTOUR LABELS ARE ROTATED
C NOND - FLAG TO TELL /CONTUN/ WHETHER OR NOT TO LABEL CONTOURS
C SIZE - HEIGHT OF CONTOUR LABELS AS PART OF AN INCH
C NDGT - NUMBER OF DIGITS IN CONTOUR LABEL
C S1D - 1-D SPECTRUM VALUES
C FREQ - ARRAY OF FREQUENCIES
C F - INPUT ARRAY FOR /CONTUN/
C FSMSK - MASK ARRAY FOR CONTUN
C FS - MASK ARRAY
      REAL S1D(500,20),FREQ(20),F(20,20),FSMSK(530),AR(3)
      INTEGER FS(18), TITLE(7)
C READ IN LENGTH OF TITLE AND TITLE OF GRAPH
      REAL LFPM
      DATA LFPM/0.06374/
      READ(5,1000) NDU
      IF(NDU .EQ. 1) GOTO 5
         LFPM=1.0
    5 CONTINUE
      READ(5,1000) NCHAR,(TITLE(I),I=1,7,1)
 1000 FORMAT(I10,7A10)
      READ (5,1010) NGRID
C     GET NUMBER OF FREQUENCIES AND LIST OF FREQUENCIES
      READ(5,100) NF,(FREQ(I),I=1,NF,1)
  100 FORMAT(I10,7F10.0,/,(8F10.0))
C     GET NUMBER OF GRID POINTS TO SKIP
      READ(5,1000) NPTSK
C     GET X AND Y DIMENSIONS OF GRID
      READ(5,1010) IGRDX,IGRDY
 1010 FORMAT(8I10)
C     START GETTING /CONTUN/ INFORMATION
C     GET MODE AND APPROPRIATE AR VALUES
      READ(5,1020) MODE,AR(1),AR(2),AR(3)
 1020 FORMAT(I10,7F10.0)
C GET LABELING FLAG AND ANGLE OF ROTATION FOR LABELS
      READ(5,1020) NOND,THETA
C     GET NUMBER OF DIGITS AND SIZE OF LABELS
      READ(5,1020) NDGT,SIZE
C     CALCULATE NUMBER OF POINTS NEEDED FROM TAPE
      READ(5,1030) SI,SJ
 1030 FORMAT(8F10.0)
C     GET 1-D SPECTRUM
      CALL READJS(1,S1D,NF,NGRID)
C     DECIDE WHERE TO START AT IN 1-D SPECTRUM
      IND = 1 + NPTSK
C     SET UP FOR PLOT ROUTINES
C     INITIALIZE THE PLOTTER
      CALL PLOTS(0,0,7)
C     MOVE PEN AWAY FROM EDGE OF PAPER
      CALL PLOT(0.5,0.5,-3)
C     GET DIMENSIONS OF PLOT
      XMAX=IGRDX/SI
      YMAX=IGRDY/SJ
C     DRAW BOX AROUND PLOT
      CALL PLOT(0.0,YMAX,2)
      CALL PLOT(XMAX,YMAX,2)
      CALL PLOT(XMAX,0.0,2)
      CALL PLOT(0.0,0.0,2)
C     SET UP NEW ORIGIN
      XINDT=1.0/(2.0*SI)
      YINDT=1.0/(2.0*SJ)
```

PROGRAM PKFRCJN

```
          ;ALL P_JI(XINDI,YINDI,-3)
C         PUT TITLE ON GRAPH
          CALL SYMBOL(J.O,YMAX,0.25,TITLE,J.J,NCHAR)
C         CALCULATE PEAK FREQUENCIES
          DO 50 L=1,IGRDY,1
            DO 5J K=1,IGRDX,1
C              SET UP FOR FINDING PEAK
               SM=S1D(IND,1)
               JM=1
C              TOP OF SEARCH LOOP
               DO 1J J=2,NF,1
C                 IS THERE A NEW MAX?
                  IF(SM.GE.S1D(IND,J)) GOTO 10
                     JM=J
                     SM=S1D(IND,J)
   10             CONTINUE
C              SEE IF PEAK IS AT ONE OF THE ENDS
               IF((JM.NE.1).AND.(JM.NE.NF)) GOTO 2D
C                 IT HAS AN ENDPOINT
                  F(K,L)=FREQ(JM)/LFPM
                  GOTO 40
   20          CONTINUE
C              GET PEAK AND FREQ ON EACH SIDE
               X1=FREQ(JM-1)
               Y1=S1D(IND,JM-1)
               X2=FREQ(JM)
               Y2=S1D(IND,JM)
               X3=FREQ(JM+1)
               Y3=S1D(IND,JM+1)
C              DETERMINE DENOMINATOR
               DENOM=X2*Y3-X3*Y2+K3*Y1-X1*Y3+X1*Y2-X2*Y1
C              IS DENOMINATOR ZERO?
               IF (DENOM .NE. 0.0) GOTO 3J
C                 DENOM HAS J, SO CURVE FIT CAN NOT BE DONE
                  F(K,L)=FREQ(JM)/LFPM
                  GOTO 40
C              GO AHEAD WITH CALCULATIONS
   30          CONTINUE
               XNUM=Y2*X3*X3-Y3*X2*X2+Y3*X1*X1-Y1*X3*X3+Y1*X2*X2-Y2*X1*X1
               F(K,L)=(-XNUM/(2*DENOM))/LFPM
   40          CONTINUE
               IND=IND+1
   50     CONTINUE
C         ZERO OUT MASK ARRAYS
          DO 98 I=1,18,1
             FS(I)=0
             FSMSK(I)=J.0
   98        CONTINUE
          DO 99 I=19,530,1
             FSMSK(I)=J.0
   99     CONTINUE
C         DUMP DATA
          WRITE(6,200)NF,(FREQ(I),I=1,NF,1)
  200     FORMAT(4H THE,I3,19H  FREQUENCIES ARE:,/,(F10.6,2X))
          WRITE(6,205) NPTSK
  205     FORMAT(1H ,I4,15H POINTS SKIPPED)
          WRITE(6,210) IGRDX,IGRDY
  210     FORMAT(9H GRID IS ,I4,3H X ,I4)
          WRITE(6,215) MODE,AR(1),AR(2),AR(3)
  215     FORMAT(5H MODE=,I2,5X,10HAR VALUES:,/,(F10.6,2X))
          WRITE(6,22J) THETA,NDGT,SIZF
  220     FORMAT(7H THETA=,F10.6,5X,5HNDGT=,I4,5X,5HSIZE=,F10.6)
          WRITE(6,225) SI,SJ
  225     FORMAT(27H NUMBER OF POINTS PER INCH:,F7.4,3H X ,F7.4)
          WRITE(6,230) XMAX,YMAX
  230     FORMAT(13H SIZE OF BOX:,F10.6,3H X ,F10.6)
C         DO CONTOUR PLOT          FINALLY!!!!!!
          CALL CONTUR(F,IGRDX,IGRDY,FS,AR,MODE,SI,SJ,THETA,NONO,SIZE,
     +          FSMSK,NDGT)
          CALL PLOT(0,J,999)
          STOP
          END
```

```
      SUBROUTINE READJS(LUN,S1D,NF,NGRID)
C
C   THIS SUBROUTINE READS 1-D SPECTRAL VALUES FOR SELECTED STATIONS
C
C   S1D - RETURNS RESULT OF SUBROUTINE
C   BUFFER - TEMPORARY INPUT BUFFER
C   LUN - LOGICAL UNIT NUMBER
C   J - FREQUENCY DO LOOP INDEX
C   L - BUFFER INDEX
C
      REAL BUFFER(501), S1D(500,NF)
      INTEGER NF,LUN,I,J,NGRID
C
C   SET FILE POINTER
      REWIND LUN
C   DO UNTIL ALL FREQUENCIES HAVE BEEN PROCESSED
      DO 30 J=1, NF, 1
         BUFFER IN (LUN,1) (BUFFER(1), BUFFER(501))
         IF (UNIT(LUN)) 10,40,40
10          DO 20 I=1,NGRID,1
               S1D(I,J)=BUFFER(I+1)
20             CONTINUE
30          CONTINUE
C   RESET FILE POINTER
      REWIND LUN
      RETURN
C   ERROR CONDITION
40 WRITE(6,50) LUN
50 FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
      STOP
      END
```

```
      SUBROUTINE CJNTUN(F,IMAX,JMAX,FS,AR,MODE,SCALEI,SCALEJ,THETA,NONO,
     1SIZE,FSMSK,NDGT)
C
C       TRACES CONTOURS THROUGH ARRAY F(IMAX,JMAX) AND PLOTS THEM
C               WHITH SCALEI,SCALEJ POINTS PR. CM. THE CONTOURS ARE
C               LABELED AT AN ANGLE THETA WITH THE X-AXES PROVIDED
C               NONO =AN INTEGER .GE.0. IF NONO.LT.0 THE CONTOURS ARE
C               NOT LABELED. SIZE IS THE SIZE OF THE LABEL.
C
C        MODE = 1 FAMILY OF CONTOURS IS TRACED WITH INTERVAL=AR(1).
C        MODE = 2 FAMILY OF APPROXIMATELY AR(1)  CONTOURS IS TRACED WITH
C                 A REASONABLY COMPUTED CONTOUR INTERVAL.
C        MODE = 3 A SINGLE CONTOUR WITH VALUE AR(1) IS TRACED.
C        MODE = 4 AR(1)=INTERVAL,AR(2)=MIN,AR(3)=MAX
C        MODE = 5 AR(1)=NR OF CONTOURS, AR(2)*FMAX=MIN,AR(3)*FMAX=MAX
C
C        IMAX COINCIDES WITH THE PAPER X-AXES. DIMENSION OF ARRAY IN X DIR
C        JMAX COINCIDES WITH THE PAPER Y-AXES. DIMENSION OF ARRAY IN Y DIR
C
C        FS  AN INTEGER ARRAY$ MUST BE GIVEN DIMENSIONS 1+IMAX*JMAX/31 IN
C                CALLING PROGRAM.
C
C        NDGT=SIGNIFICANCE,IF NEG. IS COMPUTED.
C  2 S COMPLEMENT ARITMETIC
C  EXTERNAL FUNTION LAND I,J$ IS THE BOOLEAN  I.E.LOGICAL$ AND OF TWO
C                 FULLWORD INTEGERS.
C  EXTERNAL FUNCTION LOR I,J$ IS THE BOOLEAN OR OF TWO FULLWORD INTEGERS.
C
C  SPECIFICATION STATEMENTS.
C
C       CHANGE KOS 5.1.73
      INTEGER         FS(1)
      REAL JAY,JAYZRO,F(IMAX,JMAX)
      DIMENSION MASK(31),NPT(4,3),AR(3),AC(5),ALG(5)
      DIMENSION IBEG(100),JBEG(100),CONSV(100),NINSV(100)
      INTEGER FSMSK(1)
      LOGICAL HALT,LPLOT,KLOG
      LOGICAL KLING,THING
      DATA NPT/3,4,1,2,2,1,4,3,4,3,2,1/,IFLAG/0/
      DATA AC/1.25,2.,2.5,5.,10./
      DATA ALG/.097,.3,.398,.7,1.0/
C FUNCTION DEFINITION.
      G(I,A,B)=FLOAT(I)-A/(B-A)
C SET PLOTTING PARAMETERS.
      IDFS=JMAX*IMAX
      IDFS=IDFS/31+1
      ARG2=AR(1)
      KBEG=1
      KBEGMX=0
      CSALEI=1./SCALEI
      CSALEJ=1./SCALEJ
C     IF (IOUT.NE.1) GO TO 604
C     DO 602 I=1,IMAX
C     WRITE (6,603) (F(I,J),J=1,JMAX)
C 602 CONTINUE
C 603 FORMAT(1H ,21F6.0)
C 604 CONTINUE
      IF (IFLAG.EQ.5) GO TO 4
      DO 3 I=1,31
    3 MASK(I)=2**(31-I)
C
C SET UP LOOP CONTROLLING SELECTION OF CONTOURS
C
      IFLAG=5
    4 IMAX1=IMAX-1
      JMAX1=JMAX-1
      IF (MODE.NE.3) GO TO 10
      FMIN=ARG2
      CONINT=ARG2
      NCONS=1
      GO TO 15
   10 FMAX=F(1,1)
      FMIN=FMAX
      DO 12 I=1,IMAX
      DO 12 J=1,JMAX
```

```
      FMIN=AMIN1(FMIN,F(I,J))
   12 FMAX=AMAX1(FMAX,F(I,J))
      IF (MODE.NE.5) GO TO 120
      AR(3)=AR(3)*FMAX
      AR(2)=AR(2)*FMAX
      ARG2=(AR(3)-AR(2))/AR(1)
  120 CONINT=ARG2
      IF (MODE.NE.2) GO TO 13
C IF MODE =2 SELECT A CONTOUR INTERVAL
      ALGCNT=ALOG10((FMAX-FMIN)/ARG2)
      N=ALGCNT
      ALGCNT=AMOD(ALGCNT,1.)
      IF (ALGCNT.GE.0.) GO TO 18
      N=N-1
      ALGCNT=1.+ALGCNT
   18 B=ALGCNT
      A=1.
      DO 19 I=1,5
      C=ABS(ALGCNT-ALG(I))
      IF (C.GE.B) GO TO 19
      B=C
      A=AC(I)
   19 CONTINUE
      CONINT=A*10.**N
C DETERMINE NUMBER OF CONTOURS AND MINIMUM VALUE
   13 IF (FMIN.GT.0.) FMIN=FMIN+CONINT
      FMIN=CONINT*AINT(FMIN/CONINT)
      IF (FMAX.LT.0.) FMAX=FMAX-CONINT
      FMAX=CONINT*AINT(FMAX/CONINT)
      IF (MODE.LT.4) GO TO 11
      FMIN=AMAX1(FMIN,AR(2))
      FMAX=AMIN1(FMAX,AR(3))
   11 NCONS=(FMAX-FMIN)/CONINT+1.
      NCONS=MINO(NCONS,40)
      IF(NCONS.LE.0) GO TO 7000
C DETERMINE NUMBER OF DIGITS IN LABEL
   15 DEL=1.E-4*CONINT
      IF(NDGT.GE.0) GO TO 20
      NDGT=0
   21 IF (ABS(AMOD(CONINT*10.**NDGT,1.)).LT.DEL) GO TO 20
      NDGT=NDGT+1
      IF (NDGT.LT.11) GO TO 21
C INSURE THAT NO POINT IS EXACTLY ON A CONTOUR
   20 DO 32 I=1,IMAX
      DO 32 J=1,JMAX
   32 IF (ABS(AMOD(F(I,J),CONINT)).LT.DEL) F(I,J)=1.000001*F(I,J)+2.*DEL
C START CONTOUR PLOTTING
      KONTUR=0
   16 KONTUR=KONTUR+1
      CON=FMIN+CONINT*FLOAT(KONTUR-1)
C     WRITE (6,620) CON,FMIN,CONINT,KONTUR
C 620 FORMAT(1H ,5X,  CON  ,F10.4,  FMIN  ,F10.4,  CONINT  ,F10.4,  KONT
C    1UR  ,I8)
      DO 14 MP=1,IDFS
   14 FS(MP)=0
C
C BEGIN EDGE SEARCH
C
      LPLOT=.FALSE.
      JE=0
   17 JE=JE+1
      IRET=1
   42 D=F(IMAX,JE)-CON
      C=F(IMAX,JE+1)-CON
      IF (C*D.GT.0.) GO TO 22
      IE=IMAX1
      NIN=3
      EYE=FLOAT(IMAX)
      JAY=G(JE,D,C)
      GO TO 50
   22 IF (JE.NE.JMAX1) GO TO 17
      IE=0
   23 IE=IE+1
      IRET=2
   43 A=F(IE,1)-CON
```

```
      D=F(IE+1,1)-CON
      IF (A*D.GT.0.) GO TO 24
      NIN=4
      JE=1
      EYE=G(IE,A,D)
      JAY=1.
      GO TO 50
   24 IRET=3
   41 B=F(IE,JMAX)-CON
      C=F(IE+1,JMAX)-CON
      IF(B*C.GT.0.)GO TO 26
      NIN=2
      JE=JMAX1
      EYE=G(IE,B,C)
      JAY =FLOAT(JMAX)
      GO TO 50
   26 IF(IE.NE.IMAX1)GO TO 23
C   BEGIN INTERIOR SEARCH
      IE=0
   27 IE=IE+1
      LPLOT=.TRUE.
      IF(IE.EQ.1)LPLOT=.FALSE.
      JE=0
   39 JE=JE+1
      IRET=4
   40 A=F(IE,JE)-CON
      B=F(IE,JE+1)-CON
      IF(A*B.GT.0.)GO TO  30
      NIN=1
      EYE=FLOAT(IE)
      JAY=G(JE,A,B)
      GO TO 50
   30 IF(JE.NE.JMAX1)GO TO 39
      IF(IE.NE.IMAX1)GO TO 27
      IF(KONTUR.NE.NCONS.AND.KBEG.LT.80)GO TO 16
C PURGE INITIAL POINTS BUFFER
C SEARCH STORED INITIALPOINTS BUFFER FOR ONR CLOSEST TO PRESENT PEN POSI
      KBEGMX=KBEG-1
      IRET=5
      KBMX02=KBEGMX/2+1
      K=0
   52 K=K+1
      IF(K.LT.KBMX02)GO TO 82
C RETURN IF ALL CONTOURS ARE PLOTTED
      IF(KONTUR.GE.NCONS)RETURN
      KBEG=1
      GO TO 16
   82 IRSQMN=9999999
      DO 56 KBEG=1,KBEGMX
      IF(IBEG(KBEG).EQ.0)GO TO 56
      IRSQ=(I-IBEG(KBEG))**2+(J-JBEG(KBEG))**2
      IF(IRSQ.NE.0)GO TO 53
      IF(CON.NE.CONSV(KBEG))GO TO 53
      IBEG(KBEG)=0
      GO TO 55
   53 IF(IRSQ.GE.IRSQMN)GO TO 56
      IRSQMN=IRSQ
      KBEGMN=KBEG
   56 CONTINUE
C TRACE AND PLOT CLOSEST STARTING CONTOUR
      IF(KBEGMN.LE.0.OR.KBEGMN.GT.100)GO TO 4500
      IE=IBEG(KBEGMN)
      JE=JBEG(KBEGMN)
      CON=CONSV(KBEGMN)
      NIN=NINSV(KBEGMN)
      IBEG(KBEGMN)=0
      DO 57 MP=1,IDFS
   57 FS(MP)=0
      IF(NIN.LT.1.OR.NIN.GT.4)GO TO 4500
      GO TO (40,41,42,43),NIN
   58 IF(J.EQ.0)J=1
      IF(I.EQ.0)I=1
      IF(I.EQ.IMAX)I=IMAX1
      IF(J.EQ.JMAX)J=JMAX1
      GO TO 52
```

```
CURVE FOLLOW BEGIN ROUTINE
C IF THERE IS ROOM , SAVE THE INITIAL POINT
   50 IF(KBEG.GT.100)LPLOT=.TRUE.
      IF(LPLOT)GO TO 59
      CONSV(KBEG)=CON
      IBEG(KBEG)=IE
      JBEG(KBEG)=JE
      NINSV(KBEG)=NIN
      KBEG=KBEG+1
C     IF(KBEG.GT.100)GO TO 450)
   59 IBIT=IE+IMAX*(JE-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      IF( AND(MASK(IBIT),FS(IWORD)).NE.0)GO TO (22,24,26,30),IRET
      KLING = .FALSE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING =.TRUE.
      I=IE
      J=JE
C
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(NONO.LT.0)GO TO 1000
      IF(KLING) GO TO 1100
      IF (LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C    1A,NDGT)
 1000 CONTINUE
C     IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,3)
      IF(LPLOT)CALL PLOT(EKEY,EKAY,3)
C
1100  CONTINUE
      KLING = .TRUE.
      EYEZRO=EYE
      JAYZRO=JAY
      KLOG=.FALSE.
      IPEN=2
      HALT=.FALSE.
CURVE FALLOWING ROUTINE
   60 A=F(I,J)-CON
      B=F(I,J+1)-CON
      C=F(I+1,J+1)-CON
      D=F(I+1,J)-CON
      IBIT=I+IMAX*(J-1)-1
      IWORD=IBIT/31+1
      IBIT=IBIT-31*IWORD+32
      FS(IWORD)= OR(FS(IWORD),MASK(IBIT))
      JPP=1
      IF(A*C.GE.0.)JPP=JPP+1
      IF(B*D.GE.0.)JPP=JPP+2
      IF(JPP.EQ.4)GO TO 44
   73 NOUT=NPT(NIN,JPP)
      GO TO(72,74,76,78),NOUT
   44 IF(A*B)45,46,47
   46 JPP=2
      GO TO 73
C  47 IM1=I-1
C     JM1=J-1
C     IP2=I+2
C     JP2=J+2
C     WRITE(6,45) I,J,NIN,CON,((F(IZ,JZ),IZ=IN1,IP2),JZ=JM1,JP2)
C  45 FORMAT(23H1ERROR IN CONTUR FOR I=,I2(3H) J=,I2,5H,NIN=,I1,12H,AND
C    1CONTUR=,E18.8//(4E20.8))
   47 GO TO (22,24,26,30,58),IRET
   72 EYE=FLOAT(I)
      JAY=G(J,A,B)
      IF(I.EQ.1)HALT=.TRUE.
      I=I-1
      GO TO 80
   74 EYE=G(I,B,C)
      JAY=FLOAT(J+1)
      IF(J.EQ.JMAX1)HALT=.TRUE.
      J=J+1
      GO TO 80
```

```
    76 EYE=FLOAT(I+1)
       JAY=G(J,D,C)
       IF(I.EQ.IMAX1)HALT=.TRUE.
       I=I+1
       GO TO 80
    78 EYE=G(I,A,D)
       JAY=FLOAT(J)
       IF(J.EQ.1)HALT=.TRUE.
       J=J-1
C   80 IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,IPEN)
C
    80 CONTINUE
       KLING = .TRUE.
       IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING=.FALSE.
        THING = .FALSE.
        IF(IPEN.EQ.3) THING = .TRUE.
        IPEN = 3
        IF(KLING) IPEN = 2
        IF(THING) IPEN = 3
       EKEY=EYE-1.
       EKAY=JAY-1.
       EKEY=CSALEI*EKEY
       EKAY=CSALEJ*EKAY
       IF(LPLOT)CALL PLOT(EKEY,EKAY,IPEN)
       IF(KLOG.AND.EYE.EQ.EYEZRO.AND.JAY.EQ.JAYZRO)HALT=.TRUE.
       IF(HALT)GO TO 90
       IPEN=1
       NIN=NPT(NOUT,1)
       KLOG=.TRUE.
       GO TO 60
CURVE FOLLOW END
    90 IF(NOND.LT.0)GO TO 1001
       IF(IPEN.EQ.3) GO TO 1001
       IF(LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C      IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C     1A,NDGT)
  1001 CONTINUE
       GO TO(22,24,26,30,58),IRET
  4500 WRITE(5,4501)NIN,KBEG,KBEGMX,K,IE,JE,KONTUR,CON,KBEGMN
  4501 FORMAT(4HNIN=,I15/5HKBEG=,I15/7HKBEGMX=,I15/2HK=,I15/3HIE=,I15/
      *3HJE=,I15/7HKONTUR=I15/4HCON=,E20.8/ 7HKBEGMN=,I15//)
       KBEG=1
       GO TO 15
 7000  WRITE(5,7001)
 7001  FORMAT(1H1/5X,"ERROR-NEGATIVE NO. OF CONTOURS"/1H1)
       RETURN
       END
```

F. AVERAGE FREQUENCY CONTOUR PLOTTING PROGRAM

This program produces a contour plot of average frequency with the X and Y directions representing distance. The plot is enclosed in a box and titled. The box is supposed to correspond to boundaries of the ocean that we are modeling. This program is presently set up to handle up to a 20 x 20 grid point region. It presently gets its information for only the final time step out of the SREC.

The program starts out reading in preliminary information. BUFFER is then loaded with the data to be put in the contour array. Some arrays that CONTUN needs are zeroed out. F is then loaded with normalized average frequency values. The box and title are then drawn as the paper is prepared for the contour plot. Next CONTUN is called to do the actual contour plot. Afterwords the values that were read in are written out in an attempt to let me know what data was input to the program.

INPUT LIST FOR AVERAGE FREQUENCY CONTOUR PLOTTING PROGRAM

FORMAT	VARIABLES	DESCRIPTION
I10	NDU	non-dimensional units flag; if NDU=1 then average frequency is normalized else av. freq. values are left unchanged
I10,7A10	NCHAR,TITLE	number of characters in title, title of plot
I10	NB	size of buffer
I10	NF	number of frequencies
2I10	IGRDX,IGRDY	number of grid points in X direction, in Y direction
2I10	NPTSK,IOFSET	Number of grid points to skip,number of words of information prior to first grid point value
I10,3F10.5	MODE,AR	mode flag for CONTUN, limits for CONTUN
2F10.5	SI,SJ	points per inch, x direction, y direction
I10,F10.5	NONO,THETA	if NONO=0 then contours are not labeled else contours are labeled; angle of rotation of label
I10,F10.7	NDGT,SIZE	number of decimal places to be printed in labels, size in inches of label

VARIABLE LIST FOR AVERAGE FREQUENCY CONTOUR PLOTTING PROGRAM

NDU - non-dimensional units flag, if NDU=1 then the average frequency values are divided by LFPM to make frequency non-dimensional if NDU≠1 then the average frequency values are left as they are read in

LFPM -/0.06374/ value that is divided into frequency to make it non-dimensional

NCHAR - number of characters in title

TITLE - title of plot

NB - number of words to be read in at a time, size of buffer

NF - number of frequencies

IGRDX - number of grid points in x direction

IGRDY - number of grid points in y direction

NPTSK - number of grid point values to skip in buffer

IOFSET - number of values at beginning of buffer that precede first grid point value

MODE - tells CONTUN what plotting mode to use (see CONTUN for more info)

AR - up to 3 values to be used as limits for CONTUN

SI - number of points per inch to be plotted in X direction

SJ - number of points per inch to be plotted in Y direction

NONO - labeling flag; if NONO=0 then contours are not labeled, if NONO≠0 then contours are labeled

THETA - angle of rotation of contour labels

NDGT - number of decimal places used in labels

SIZE - size in inches of labels

BUFFER - input buffer, temporarily holds data from tape before it is put into F to be plotted

FS - array required by CONTUN, must be set to 0

FSMSK - an array required by CONTUN, must be set to 0

IND - index into BUFFER, starts out as 1+IOFSET+NPTSK

XMAX - length of X axis in inches

YMAX - length of Y axis in inches

XINDT - new x coordinate for origin

YINDT - new Y coordinate for origin

READS- this subroutine reads an SREC

PARAMETERS-

LUN - logical unit number, device number of tape with SREC
BUFFER - input buffer that is to receive the SREC data
NB - number of words to be read in
NF - number of frequencies

LOCAL VARIABLES-

I - do loop index
DUM - temp location used to read into while skipping over records

TEXT-

This subroutine spaces the tape forward to the SREC and then loads it into BUFFER. The number of words read in depends on NB.

```
      PROGRAM AVFQ(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C
C THIS ROUTINE PLOTS AVERAGE FREQUENCY FROM CYCLE TAPE DATA.
C
C AR - INTERVAL VALUES FOR /CONTUN/
C F - INPUT ARRAY FOR /CONTUN/
C FS - MASK ARRAY
C FSMSK - MASK ARRAY FOR CONTUN
C IGRDX - NUMBER OF GRID POINTS IN X DIRECTION
C IGRDY - NUMBER OF GRID POINTS IN Y DIRECTION
C MODE - SPECIFIES MODE FOR /CONTUN/
C NDGT - NUMBER OF DIGITS IN CONTOUR LABEL
C NF - NUMBER OF FREQUENCIES
C NONO - FLAG TO TELL /CONTUN/ WHETHER OR NOT TO LABEL CONTOURS
C NPTSK - NUMBER OF GRID POINTS TO SKIP
C SI - NUMBER OF POINTS PER INCH IN X DIRECTION
C SJ - NUMBER OF POINTS PER INCH IN Y DIRECTION
C SIZE - HEIGHT OF CONTOUR LABELS AS PART OF AN INCH
C THETA - ANGLE THAT CONTOUR LABELS ARE ROTATED
      REAL F(23,20),BUFFER(4511)
      REAL AR(3),SI,SJ,THETA,SIZE
      REAL FSMSK(530),XMAX,YMAX,XINDT,YINDT
      INTEGER NB,NF,IGRDX,IGRDY,IOFSET,NPTSK,MODE,FS(18),TITLE(7)
      REAL LFPM
      DATA LFPM/0.06374/
      READ(5,1000) NDU
      IF(NDU .EQ. 1) GOTO 5
         LFPM=1.0
    5 CONTINUE
C READ IN LENGTH OF TITLE AND TITLE OF GRAPH
      READ(5,1000) NCHAR,(TITLE(I),I=1,7,1)
C     GET SIZE OF BUFFER
      READ(5,1010) NB
C     GET NUMBER OF FRUENCIES
      READ(5,1010) NF
C     GET X AND Y DIMENSIONS OF GRID
      READ(5,1010) IGRDX,IGRDY
C     GET NUMBER OF GRID POINTS TO SKIP AND IOFSET
      READ(5,1010) NPTSK,IOFSET
C     GET MODE AND AR VALUES
      READ(5,1020) MODE,AR(1),AR(2),AR(3)
C     POINTS PER INCH
      READ(5,1030) SI,SJ
C GET LABELING FLAG AND ANGLE OF ROTATION FOR LABELS
      READ(5,1020) NONO,THETA
C     GET NUMBER OF DIGITS AND SIZE OF LABELS
      READ(5,1020) NDGT,SIZE
 1000 FORMAT(I10,7A10)
 1010 FORMAT(8I10)
 1020 FORMAT(I10,7F10.0)
 1030 FORMAT(8F10.0)
C     LOAD BUFFER
      CALL READS(1,BUFFER,NB,NF)
      DO 98 I=1,18,1
         FS(I)=0
         FSMSK(I)=0.0
   98    CONTINUE
      DO 99 I=19,530,1
         FSMSK(I)=0.0
   99 CONTINUE
C     ADJUST BUFFER INDEX
      IND = IOFSET + NPTSK + 1
C     LOAD F
      DO 10 J=1,IGRDY,1
         DO 10 I=1,IGRDX,1
            F(I,J)=BUFFER(IND)/LFPM
            IND = IND + 1
   10    CONTINUE
C     GET DIMENSIONS OF PLOT
      XMAX=IGRDX/SI
      YMAX=IGRDY/SJ
C     INITIALIZE THE PLOTTER
      CALL PLOTS(0,0,7)
C     MOVE PEN AWAY FROM EDGE OF PAPER
      CALL PLOT(0.5,0.5,-3)
```

PROGRAM AVFO

```
C     DRAW BOX AROUND PLOT
      CALL PLOT(0.0,YMAX,2)
      CALL PLOT(XMAX,YMAX,2)
      CALL PLOT(XMAX,0.0,2)
      CALL PLOT(0.0,0.0,2)
C     SET UP NEW ORIGIN
      XINDT=1.0/(2.0*SI)
      YINDT=1.0/(2.0*SJ)
      CALL PLOT(XINDT,YINDT,-3)
C     PUT TITLE ON GRAPH
      CALL SYMBOL(0.0,YMAX,0.25,TITLE,0.0,NCHAR)
C     DO CONTOUR PLOT            FINALLY))))))
      CALL CONTUN(F,IGRDX,IGRDY,FS,AR,MODE,SI,SJ,THETA,NONO,SIZE,
     +            FSMSK,NDGT)
      CALL PLOT(0,0,999)
C     DUMP DATA
      WRITE(6,220) THETA,NDGT,SIZE
  220 FORMAT(7H THETA=,F10.6,5X,5HNDGT=,I4,5X,5HSIZE=,F10.6)
      WRITE(6,230) XMAX,YMAX
  230 FORMAT(13H SIZE OF BOX:,F10.6,3H X ,F10.6)
      WRITE(6,300) NB
  300 FORMAT(4H NB=,I4)
      WRITE(6,301) NF
  301 FORMAT(4H NF=,I4)
      WRITE(6,302) IGRDX,IGRDY
  302 FORMAT(9H GRID IS ,I4,3H X ,I4)
      STOP
      END
```

SUBROUTINE READS

```
      SUBROUTINE READS (LUN, BUFFER, NB, NF)
C
C     SUBROUTINE TO READ SUMMARY RECORD - -
C          SUMS OVER FREQUENCY AND DIRECTION.
C
      REAL BUFFER(NB)
C
      REWIND LUN
C     SKIP TO SREC.
      DO 10 I=1,NF
      BUFFER IN (LUN,1) (DUM,DUM)
C
      IF (UNIT (LUN)) 10, 50, 50
C
 10   CONTINUE
C     READ SREC.
      BUFFER IN (LUN,1) (BUFFER(1),BUFFER(NB))
C
      IF (UNIT (LUN)) 20, 50, 50
C
 20   CONTINUE
      REWIND LUN
C
      RETURN
C
 50   WRITE (6,1000) LUN
 1000 FORMAT (32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
C
      STOP
C
      END
```

```
      SUBROUTINE CONTUN(F,IMAX,JMAX,FS,AR,MODE,SCALEI,SCALEJ,THETA,NONO,
     1SIZE,FSMSK,NDGT)
C
C      TRACES CONTOURS THROUGH ARRAY F(IMAX,JMAX) AND PLOTS THEM
C             WHITH SCALEI,SCALEJ POINTS PR. CM. THE CONTOURS ARE
C             LABELED AT AN ANGLE THETA WITH THE X-AXES PROVIDED
C             NONO =AN INTEGER .GE.0. IF NONO.LT.0 THE CONTOURS ARE
C             NOT LABELED. SIZE IS THE SIZE OF THE LABEL.
C
C       MODE = 1 FAMILY OF CONTOURS IS TRACED WITH INTERVAL=AR(1).
C       MODE = 2 FAMILY OF APPROXIMATELY AR(1)  CONTOURS IS TRACED WITH
C                A REASONABLY COMPUTED CONTOUR INTERVAL.
C       MODE = 3 A SINGLE CONTOUR WITH VALUE AR(1) IS TRACED.
C       MODE = 4 AR(1)=INTERVAL,AR(2)=MIN,AR(3)=MAX
C       MODE = 5 AR(1)=NR OF CONTOURS, AR(2)*FMAX=MIN,AR(3)*FMAX=MAX
C
C       IMAX COINCIDES WITH THE PAPER X-AXES. DIMENSION OF ARRAY IN X DIR
C       JMAX COINCIDES WITH THE PAPER Y-AXES. DIMENSION OF ARRAY IN Y DIR
C
C       FS  AN INTEGER ARRAYS MUST BE GIVEN DIMENSIONS 1+IMAX*JMAX/31 IN
C             CALLING PROGRAM.
C
C       NDGT=SIGNIFICANCE,IF NEG. IS COMPUTED.
C 2 S COMPLEMENT ARITMETIC
C EXTERNAL FUNTION LAND I,J$ IS THE BOOLEAN  I.E.LOGICAL$ AND OF TWO
C               FULLWORD INTEGERS.
C EXTERNAL FUNCTION LOR I,J$ IS THE BOOLEAN OR OF TWO FULLWORD INTEGERS.
C
C SPECIFICATION STATEMENTS.
C
C      CHANGE KOS 5.1.73
      INTEGER         FS(1)
      REAL JAY,JAYZRO,F(IMAX,JMAX)
      DIMENSION MASK(31),NPT(4,3),AR(3),AC(5),ALG(5)
      DIMENSION IBEG(100),JBEG(100),CONSV(100),NINSV(100)
      INTEGER FSMSK(1)
      LOGICAL HALT,LPLOT,KLOG
      LOGICAL KLING,THING
      DATA NPT/3,4,1,2,2,1,4,3,4,3,2,1/,IFLAG/0/
      DATA AC/1.25,2.,2.5,5.,10./
      DATA ALG/.097,.3,.398,.7,1.0/
C FUNCTION DEFINITION.
      G(I,A,B)=FLOAT(I)-A/(B-A)
C SET PLOTTING PARAMETERS.
      IDFS=JMAK*IMAX
      IDFS=IDFS/31+1
      ARG2=AR(1)
      KBEG=1
      KBEGMX=0
      CSALEI=1./SCALEI
      CSALEJ=1./SCALEJ
C     IF (IOUT.NE.1) GO TO 604
C     DO 602 I=1,IMAX
C     WRITE (6,603) (F(I,J),J=1,JMAX)
C 602 CONTINUE
C 603 FORMAT(1H ,21F6.0)
C 604 CONTINUE
      IF (IFLAG.EQ.5) GO TO 4
      DO 3 I=1,31
    3 MASK(I)=2**(31-I)
C
C SET UP LOOP CONTROLLING SELECTION OF CONTOURS
C
      IFLAG=5
    4 IMAX1=IMAX-1
      JMAX1=JMAX-1
      IF (MODE.NE.3) GO TO 10
      FMIN=ARG2
      CONINT=ARG2
      NCONS=1
      GO TO 15
   10 FMAX=F(1,1)
      FMIN=FMAK
      DO 12 I=1,IMAX
      DO 12 J=1,JMAX
```

```
      FMIN=AMIN1(FMIN,F(I,J))
   12 FMAX=AMAX1(FMAX,F(I,J))
      IF (MODE.NE.5) GO TO 120
      AR(3)=AR(3)*FMAX
      AR(2)=AR(2)*FMAX
      ARG2=(AR(3)-AR(2))/AR(1)
  120 CONINT=ARG2
      IF (MODE.NE.2) GO TO 13
C IF MODE =2 SELECT A CONTOUR INTERVAL
      ALGCNT=ALOG10((FMAX-FMIN)/ARG2)
      N=ALGCNT
      ALGCNT=AMOD(ALGCNT,1.)
      IF (ALGCNT.GE.0.) GO TO 18
      N=N-1
      ALGCNT=1.+ALGCNT
   18 B=ALGCNT
      A=1.
      DO 19 I=1,5
      C=ABS(ALGCNT-ALG(I))
      IF (C.GE.B) GO TO 19
      B=C
      A=AC(I)
   19 CONTINUE
      CONINT=A*10.**N
C DETERMINE NUMBER OF CONTOURS AND MINIMUM VALUE
   13 IF (FMIN.GT.0.) FMIN=FMIN+CONINT
      FMIN=CONINT*AINT(FMIN/CONINT)
      IF (FMAX.LT.0.) FMAX=FMAX-CONINT
      FMAX=CONINT*AINT(FMAX/CONINT)
      IF (MODE.LT.4) GO TO 11
      FMIN=AMAX1(FMIN,AR(2))
      FMAX=AMIN1(FMAX,AR(3))
   11 NCONS=(FMAX-FMIN)/CONINT+1.
      NCONS=MINO(NCONS,40)
      IF(NCONS.LE.0) GO TO 7000
C DETERMINE NUMBER OF DIGITS IN LABEL
   15 DEL=1.E-4*CONINT
      IF(NDGT.GE.0) GO TO 20
      NDGT=0
   21 IF (ABS(AMOD(CONINT*10.**NDGT,1.)).LT.DEL) GO TO 20
      NDGT=NDGT+1
      IF (NDGT.LT.11) GO TO 21
C INSURE THAT NO POINT IS EXACTLY ON A CONTOUR
   20 DO 32 I=1,IMAX
      DO 32 J=1,JMAX
   32 IF (ABS(AMOD(F(I,J),CONINT)).LT.DEL) F(I,J)=1.000001*F(I,J)+2.*DE
C START CONTOUR PLOTTING
      KONTUR=0
   16 KONTUR=KONTUR+1
      CON=FMIN+CONINT*FLOAT(KONTUR-1)
C     WRITE (6,620) CON,FMIN,CONINT,KONTUR
C 620 FORMAT(1H ,5X,  CON   ,F10.4,  FMIN   ,F10.4,  CONINT   ,F10.4,  KONT
C    1UR   ,I8)
      DO 14 MP=1,IDFS
   14 FS(MP)=0
C
C BEGIN EDGE SEARCH
C
      LPLOT=.FALSE.
      JE=0
   17 JE=JE+1
      IRET=1
   42 D=F(IMAX,JE)-CON
      C=F(IMAX,JE+1)-CON
      IF (C*D.GT.0.) GO TO 22
      IE=IMAX1
      NIN=3
      EYE=FLOAT(IMAX)
      JAY=G(JE,D,C)
      GO TO 50
   22 IF (JE.NE.JMAX1) GO TO 17
      IE=0
   23 IE=IE+1
      IRET=2
   43 D=F(IE,1)-CON
```

```
      D=F(IE+1,1)-CON
      IF (A*D.GT.0.) GO TO 24
      NIN=4
      JE=1
      EYE=G(IE,A,D)
      JAY=1.
      GO TO 50
   24 IRET=3
   41 B=F(IE,JMAX)-CON
      C=F(IE+1,JMAX)-CON
      IF(B*C.GT.0.)GO TO 26
      NIN=2
      JE=JMAX1
      EYE=G(IE,B,C)
      JAY =FLOAT(JMAX)
      GO TO 50
   26 IF(IE.NE.IMAX1)GO TO 23
C   BEGIN INTERIOR SEARCH
      IE=0
   27 IE=IE+1
      LPLOT=.TRUE.
      IF(IE.EQ.1)LPLOT=.FALSE.
      JE=0
   39 JE=JE+1
      IRET=4
   40 A=F(IE,JE)-CON
      B=F(IE,JE+1)-CON
      IF(A*B.GT.0.)GO TO  30
      NIN=1
      EYE=FLOAT(IE)
      JAY=G(JE,A,B)
      GO TO 50
   30 IF(JE.NE.JMAX1)GO TO 39
      IF(IE.NE.IMAX1)GO TO 27
      IF(KONTUR.NE.NCONS.AND.KBEG.LT.80)GO TO 16
C PURGE INITIAL POINTS BUFFER
C SEARKH STORED INITIALPOINTS BUFFER FOR ONR CLOSEST TO PRESENT PEN POSI
      KBEGMX=KBEG-1
      IRET=5
      KBMX02=KBEGMX/2+1
      K=0
   52 K=K+1
      IF(K.LT.KBMX02)GO TO 82
C RETURN IF ALL CONTOURS ARE PLOTTED
      IF(KONTUR.GE.NCONS)RETURN
      KBEG=1
      GO TO 15
   82 IRSQMN=9999999
      DO 56 KBEG=1,KBEGMX
      IF(IBEG(KBEG).EQ.0)GO TO 56
      IRSQ=(I-IBEG(KBEG))**2+(J-JBEG(KBEG))**2
      IF(IRSQ.NE.0)GO TO 53
      IF(CON.NE.CONSV(KBEG))GO TO 53
      IBEG(KBEG)=0
      GO TO 55
   53 IF(IRSQ.GE.IRSQMN)GO TO 56
      IRSQMN=IRSQ
      KBEGMN=KBEG
   56 CONTINUE
C TRACE AND PLOT CLOSEST STARTING CONTOUR
      IF(KBEGMN.LE.0.OR.KBEGMN.GT.100)GO TO 4500
      IE=IBEG(KBEGMN)
      JE=JBEG(KBEGMN)
      CON=CONSV(KBEGMN)
      NIN=NINSV(KBEGMN)
      IBEG(KBEGMN)=0
      DO 57 MP=1,IDFS
   57 FS(MP)=0
      IF(NIN.LT.1.OR.NIN.GT.4)GO TO 4500
      GO TO (40,41,42,43),NIN
   58 IF(J.EQ.0)J=1
      IF(I.EQ.0)I=1
      IF(I.EQ.IMAX)I=IMAX1
      IF(J.EQ.JMAX)J=JMAX1
      GO TO 52
```

```
CURVE FOLLOW BEGIN ROUTINE
C IF THERE IS ROOM , SAVE THE INITIAL POINT
    50 IF(KBEG.GT.100)LPLOT=.TRUE.
       IF(LPLOT)GO TO 59
       CONSV(KBEG)=CON
       IBEG(KBEG)=IE
       JBEG(KBEG)=JE
       NINSV(KBEG)=NIN
       KBEG=KBEG+1
C      IF(KBEG.GT.100)GO TO 450)
    59 IBIT=IE+IMAX*(JE-1)-1
       IWORD=IBIT/31+1
       IBIT=IBIT-31*IWORD+32
       IF( AND(MASK(IBIT),FS(IWORD)).NE.0)GO TO (22,24,26,30),IRET
       KLING = .FALSE.
       IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING =.TRUE.
       I=IE
       J=JE
C
       EKEY=EYE-1.
       EKAY=JAY-1.
       EKEY=CSALEI*EKEY
       EKAY=CSALEJ*EKAY
       IF(NONO.LT.0)GO TO 1000
       IF(KLING) GO TO 1100
       IF (LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C      IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C     1A,NDGT)
 1000  CONTINUE
C      IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,3)
       IF(LPLOT)CALL PLOT(EKEY,EKAY,3)
C
1100   CONTINUE
       KLING = .TRUE.
       EYEZRO=EYE
       JAYZRO=JAY
       KLOG=.FALSE.
       IPEN=2
       HALT=.FALSE.
CURVE FALLOWING ROUTINE
    60 A=F(I,J)-CON
       B=F(I,J+1)-CON
       C=F(I+1,J+1)-CON
       D=F(I+1,J)-CON
       IBIT=I+IMAX*(J-1)-1
       IWORD=IBIT/31+1
       IBIT=IBIT-31*IWORD+32
       FS(IWORD)= OR(FS(IWORD),MASK(IBIT))
       JPP=1
       IF(A*C.GE.0.)JPP=JPP+1
       IF(B*D.GE.0.)JPP=JPP+2
       IF(JPP.EQ.4)GO TO 44
    73 NOUT=NPT(NIN,JPP)
       GO TO(72,74,76,78),NOUT
    44 IF(A*B)45,46,47
    46 JPP=2
       GO TO 73
C   47 IM1=I-1
C      JM1=J-1
C      IP2=I+2
C      JP2=J+2
C     WRITE(6,45) I,J,NIN,CON,((F(IZ,JZ),IZ=IM1,IP2),JZ=JM1,JP2)
C   45 FORMAT(23H1ERROR IN CONTUR FOR I=,I2(3H) J=,I2,5H,NIN=,I1,12H,AND
C     1CONTUR=,E18.8//(4E20.8))
    47 GO TO (22,24,26,30,58),IRET
    72 EYE=FLOAT(I)
       JAY=G(J,A,B)
       IF(I.EQ.1)HALT=.TRUE.
       I=I-1
       GO TO 80
    74 EYE=G(I,B,C)
       JAY=FLOAT(J+1)
       IF(J.EQ.JMAX1)HALT=.TRUE.
       J=J+
       GO TO 80
```

```
      76 EYE=FLOAT(I+1)
         JAY=G(J,D,C)
         IF(I.EQ.IMAXL)HALT=.TRUE.
         I=I+1
         GO TO 80
      78 EYE=G(I,A,D)
         JAY=FLOAT(J)
         IF(J.EQ.1)HALT=.TRUE.
         J=J-1
C     80 IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,IPEN)
C
      80 CONTINUE
         KLING = .TRUE.
         IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING=.FALSE.
          THING = .FALSE.
          IF(IPEN.EQ.3) THING = .TRUE.
          IPEN = 3
          IF(KLING) IPEN = 2
          IF(THING) IPEN = 3
         EKEY=EYE-1.
         EKAY=JAY-1.
         EKEY=CSALEI*EKEY
         EKAY=CSALEJ*EKAY
         IF(LPLOT)CALL PLOT(EKEY,EKAY,IPEN)
         IF(KLOG.AND.EYE.EQ.EYEZRO.AND.JAY.EQ.JAYZRO)HALT=.TRUE.
         IF(HALT)GO TO 90
         IPEN=1
         NIN=NPT(NOUT,1)
         KLOG=.TRUE.
         GO TO 60
CURVE FOLLOW END
      90 IF(NONO.LT.0)GO TO 1001
         IF(IPEN.EQ.3) GO TO 1001
         IF(LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
C        IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
C       1A,NDGT)
    1001 CONTINUE
         GO TO(22,24,26,30,58),IRET
    4500 WRITE(5,4501)NIN,KBEG,KBEGMX,K,IE,JE,KONTUR,CON,KBEGMN
    4501 FORMAT(4HNIN=,I15/5HKBEG=,I15/7HKBEGMX=,I15/2HK=,I15/3HIE=,I15/
        *3HJE=,I15/7HKONTUR=I15/4HCON=,E20.8/ 7HKBEGMN=,I15//)
         KBEG=1
         GO TO 15
7000     WRITE(5,7001)
7001     FORMAT(1H1/5X,"ERROR-NEGATIVE NO. OF CONTOURS"/1H1)
         RETURN
         END
```

G. <u>2-D SPECTRUM CONTOUR</u>

This program produces a contour plot of the 2-D Spectrum with frequency in the X direction and Theta in the Y direction. The frequencies are interpolated to make equal distances in the X direction reflect equal changes in frequency. The range is from 0 to 0.229 HZ. The Thetas are plotted side by side although the sampling is not equal. The list of directions, from top to bottom, is: 180, 135, 90, 75, 60, 45, 30, 15, 0, -15, -30, -45, -60, -75, -90, -135, and -180.

The program works into several different sections. First the program goes about reading in information about the plot and the data. Next the program sets up all the preliminary data for CONTUN. The program then calls FNDREC to get the tape positioned at the appropriate time step. The program then loads the 2-D array F with the available information. Next comes a section of code to interpolate for missing frequencies. Just prior to calling CONTUN which actually does the contour plot the program draws a box around the region where the plot will be made and then titles it.

INPUT LIST FOR 2-D SPECTRUM PROGRAM

FORMAT	VARNAME	DESCRIPTION
I10	NCHAR	number of characters in title
7A10	TITLE	title of graph
I10	GRDPT	grid point number
I10	TIMSTP	time step number
I10	INF	input number of frequencies
8I10	IFREQ(i),i=1,INF	list of frequency indicies
I10	IND	input number of directions
8I10	FIND(i),i=1,IND	list of direction indices
I10	NGRID	number of grid points
F10.5	LFRAX	length of frequency axis in inches
F10.5	LDIRAX	length of direction axis in inches
I10,3F10.5	MODE,AR	mode flag followed by appropriate AR values
I10,F10.5	NONO,Theta	NONO=1 then contours are labeled, NONO=0 then contours are not labeled; THETA is angle of rotation of label
I10,F10.5	NDGT,SIZE	number of digits in label,size in inches of label

2-D SPECTRUM CONTOUR PLOT PROGRAM

VARIABLES -

BUF(300) - input buffer
LFRAX - length of frequency axis in inches
LDIRAX - length of direction axis in inches
AR(3) - array passed to CONTUN
if MODE=1 then contour interval is AR(1)
if MODE=2 then approximately AR(1) contours will be traced
if MODE=3 then a single contour with value AR(1) will be drawn
if MODE=4 then AR(1)=interval, AR(2)=min, AR(3)=max
if MODE=5 then AR(1)=num of contours, AR(2)=min/fmax, AR(3)=max/fmax
MODE - tells CONTUN what type of plotting mode to use
THETA - angle of rotation that is used when writing the labels on the contours
SIZE - size of label in fraction of an inch.
F(42,30) - the array that CONTUN uses for the values to create the contours
FIND(30) - list of indices, FIND(n) yields the direction index for the nth direction, the index is used as a subscript into F
IFREQ(30) - list of indices, IFREQ(j) yields the frequency index for the jth frequency, F(IFREQ(j),FIND(n)) would tell where to put a given frequency-index value from the tape
NCHAR - number of characters in title of graph
TITLE - title of graph
GRDPT - grid point number that is to be plotted
TIMSTP - time step number that is to be plotted
INF - input number of frequencies, later becomes total number of frequencies after interpolation
IND - input number of directions, becomes final number of directions after 180 is copied to -180
NGRID - number of grid points (from waveset, must know so tape can be positioned)
NONO - if NONO=1 then contours are labeled, if NONO=0 then contours are not labeled
NDCT - number of decimal places to be displayed in contour labels
FS(15) - mask array used by CONTUN must be set to 0
FSMSK(15) - mask array used by CONTUN, must be set to 0
I1 - tells which half of the word the data i want is in
I2 - tells which word in BUF contains the data i want
EOB - tells me how many words to read in at a time
J1 - IFREQ(j)
K1 - FIND(k)
XPPI - number of points per inch in x direction
YPPI - number of points per inch in y direction

ISPLIT- this subroutine(function) unpacks an integer word

PARAMETERS-

IND - flag indicating which half the word to use
IWORD - the integer word containing the two values

TEXT-

This function returns the value stored in half of the integer word. This unpacking is necessary because two values have been compressed into one integer word and written on the tape. This function will choose either half of a sixty-bit integer word(IWORD) depending on the value in IND.

RSPLIT- this function returns a value stored in either the first or second half of a real variable

PARAMETERS-

IND - indicates whether you want value in first or second half of word

WORD - the real word that contains the two packed values

TEXT-

This function unpacks a value from the first or second half of a sixty bit real word. It is needed because data is packed on the tape as two values per word.

FNDREC- this subroutine finds the proper time step on a tape

PARAMETERS-

NSTEP - time step number to be searched for
LUN - logical unit number of tape to be searched
ND - number of directions on tape
NF - number of frequencies on tape

LOCAL VARIABLES-

IBUF - temporary input buffer
J - do loop index
K - do loop index

TEXT-

This subroutine spaces the tape forward until the appropriate time step is found according to a header record. It leaves the tape ready to read the data records for that time step.

```
      PROGRAM S2DCON(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2,TAPE7)
      REAL BUF(300),LFRAX,LDIRAX,AR(3),THETA,SIZE,T
      REAL MAX
      REAL F(32,25)
      INTEGER FIND(30)
      INTEGER IFREQ(30)
      INTEGER NCHAR,TITLE(7),GROPT,TIMSTP,INF,IND,NGRID,INF,IND
      INTEGER IDIR(30),MODE,NONO,NDGT,FS(30),FSMASK(30),I1
      INTEGER I,J1,J2,K1,K2
C NCHAR - NUMBER OF CHARACTERS IN TITLE
C TITLE - TITLE OF GRAPH
C GROPT - GRID POINT NUMBER TO BE PLOTTED
C TIMSTP - TIME STEP NUMBER TO BE PLOTTED
C NGRID - TOTAL NUMBER OF GRID POINTS
C BUF - INPUT BUFFER
C INF - NUMBER OF FREQUENCIES TO BE PLOTTED
C IFREQ - ARRAY OF INDICES OF  OF FREQUENCIES TO BE PLOTTED
C IND - NUMBER OF DIRECTIONS TO BE PLOTTED
C IDIR- ARRAY OF DIRECTION INDICES TO BE PLOTTED
C LFRAX - LENGTH OF FREQUENCY AXIS IN INCHES
C LDIRAX - LENGTH OF DIRECTION AXIS IN INCHES
C MODE - MODE FLAG FOR CONTUN
C AR - ARRAY OF MIN,MAX,INTERVAL INFORMATION FOR CONTUN
C THETA - ANGLE OF ROTATION OF LABELS ON CONTOURS
C NONO - FLAG THAT TELLS CONTUN WHETHER OR NOT TO LABEL THE CONTOURS
C SIZE - HEIGHT OF LABELS IN INCHES
C NDGT - NUMBER OF SIGNIFICANT DIGITS IN CONTOUR LABELS
C F - ARRAY OF VALUES TO BE PLOTTED
C FS - MASK ARRAY
C FSMASK - MASK ARRAY
      BFPM=865400.0
      READ(5,1000) NCHAR,TITLE
 1000 FORMAT(I10,/,8A10)
      WRITE(6,2000) NCHAR,TITLE
 2000 FORMAT(31H NUMBER OF CHARACTERS IN TITLE:,I2,/,
     +  11H TITLE IS :,8A10)
      READ(5,1001) GROPT
 1001 FORMAT(8I10)
      WRITE(6,2001) GROPT
 2001 FORMAT(25H SELECTED GRID POINT IS :,I5)
      READ(5,1001) TIMSTP
      WRITE(6,2013) TIMSTP
 2013 FORMAT(12H TIME STEP :,I4)
      READ(5,1001) INF
      WRITE(6,2002)  INF
 2002 FORMAT(33H INPUT NUMBER OF FREQUENCIES :,I4)
      READ(5,1001) (IFREQ(I),I=1,INF,1)
      WRITE(6,2003) (IFREQ(I),I=1,INF,1)
 2003 FORMAT(23H FREQUENCY INDEX LIST :,/,(1H ,8I10))
      READ(5,1001) IND
      WRITE(6,2004) IND
 2004 FORMAT(5H IND=,I5)
      READ(5,1001) (FIND(I),I=1,IND,1)
      WRITE(6,2014) (FIND(I),I=1,IND,1)
 2014 FORMAT(5H FIND:,(/,1H ,8I10))
      READ(5,1001) NGRID
      WRITE(6,2005) NGRID
 2005 FORMAT(24H NUMBER OF GRID POINTS :,I4)
      READ(5,1002) LFRAX
 1002 FORMAT(8F10.5)
      READ(5,1002) LDIRAX
      READ(5,1003) MODE,AR
 1003 FORMAT(I10,7F10.5)
      WRITE(6,2010) MODE,AR
 2010 FORMAT(5H MODE=,I4,/,7H AR(1)=,F10.6,/,7H AR(2)=,F10.6,/,
     +  7H AR(3)=,F10.6)
      READ(5,1003) NONO,THETA
      WRITE(6,2011) NONO , THETA
 2011 FORMAT(13H LABEL FLAG :,I4,/,20H ANGLE OF ROTATION :,F10.6)
      READ(5,1003) NDGT,SIZE
      WRITE(6,2012) NDGT,SIZE
 2012 FORMAT(17H LABEL WILL HAVE ,I3,19H SIGNIFICANT DIGITS,/,
     +  24H AND EACH DIGIT WILL BE F10.6,12H INCHES TALL)
      DO 10 I=1,30,1
         FS(I)=0
```

```
            FSMASK(I)=F
   10       CONTINUE
         DO 12 I=1,32,1
            DO 11 J=1,25,1
               F(I,J)=0.0
   11          CONTINUE
   12       CONTINUE
C FIND CORRECT TIME STEP
         REWIND 2
         CALL FNDREC(FINDSTP,2,IND,20)
C  SET UP INITIAL VALUES
         I1=FIND(GROPT+1,2)+1
         I2=(GROPT+9)/2
         IDL=(4*NG-1)/2
         MAX=0.0
         DO 60  J=1,INF,1
            DO 50 K=1,IND,1
               BUFFER IN (2,1) (BUF(1),BUF(EOB))
               IF(UNIT(2)) 30,99,99
   30          CONTINUE
               J1=IFREQ(J)
               K1=FIND(K)
               F(J1,K1)=RSPLIT(I1,BUF(I2))
         MAX=AMAX1(MAX,F(J1,K1))
   50          CONTINUE
   60       CONTINUE
         DO 100 K=1,IND,1
            I=FIND(K)
            DO 90 J=9,23,2
            F(J,I)=(F(J-1,I)+F(J+1,I))/2.0
   90          CONTINUE
            T=(F(28,I)-F(24,I))/4.0
            F(25,I)=F(24,I)+T
            F(26,I)=F(25,I)+T
            F(27,I)=F(26,I)+T
            T=(F(32,I)-F(28,I))/4.0
            F(29,I)=F(28,I)+T
            F(30,I)=F(29,I)+T
            F(31,I)=F(30,I)+T
  100       CONTINUE
         WRITE(6,198) MAX
  198 FORMAT(1H MAX IS  ,F15.7)
         INF=32
         DO 79 I=1,INF,1
            DO 79 K=1,IND,1
               J=FIND(K)
               F(I,J)=F(I,J)/MAX
   79          CONTINUE
         IND=25
         DO 80 I=1,INF,1
            F(I,IND)=F(I,1)
            DO 80 J=1,IND,1
          WRITE(6,199) I,J,F(I,J)*MAX,F(I,J)
  199 FORMAT(1H ,I5,5X,I5,5X,F15.7,5X,F15.7)
   80       CONTINUE
C   NOW PREPARE FOR PLOT
         CALL PLOTS(0,0,7)
C  MOVE PEN AWAY FROM EDGE OF PAPER
         CALL PLOT(0.5,0.5,-3)
C DRAW BOX AROUND PLOT
         LFRAX=5.102
         LOIRAX=5.3
         PX=1.772
         PY=1.0
         XPPI=(INF-1)/LFRAX
         YPPI=(IND-1)/LOIRAX
         BXL=7.37
         BYL=6.5
         CALL PLOT(0.0,BYL,2)
         CALL PLOT(BXL,BYL,2)
         CALL PLOT(BXL,0.0,2)
         CALL PLOT(0.0,0.0,2)
         CALL PLOT(PX,PY,-3)
C   PUT TITLE ON GRAPH
         CALL SYMBOL(0.  ,BYL    ,0.25,TITLE,0.0,NCHAR)
```

PROGRAM S2DCON

```
      CALL CONTOUR(F,INF,IND,FS,AR,MODE,XPPI,YPPI,THETA,NONO,SIZE,FSMAS,
     + NOUT)
      CALL PLOT(0,0,999)
      STOP
   99 CONTINUE
      WRITE(6,999)
  999 FORMAT(12H INPUT ERROR)
      STOP
      END
```

FUNCTION RSPLIT

```
      FUNCTION RSPLIT(IND,WORD)
C
      INTEGER IND
      REAL WORD,TEMP
      GO TO (100,110) , IND
  100 RSPLIT = WORD .AND. 77777777770000000000B
      RETURN
  110 TEMP = WORD .AND. 00000000007777777777B
      RSPLIT = SHIFT(TEMP,30)
      RETURN
      END
```

FUNCTION ISPLIT

```
      FUNCTION ISPLIT(IND,IWORD)
C
      INTEGER IND,IWORD,ITEMP
      GOTO (100,110),IND
  100 ITEMP = IWORD .AND. 77777777770000000000B
      ISPLIT = SHIFT(ITEMP,-30)
      RETURN
  110 ITEMP = IWORD .AND. 00000000007777777777B
      ITEMP = SHIFT(ITEMP,30)
      ISPLIT = SHIFT(ITEMP,-30)
      RETURN
      END
```

```
      SUBROUTINE FNDREC(NSTEP,LUN,ND,NF)
C
C  THIS SUBROUTINE SPACES TAPE FORWARD UNTIL PROPER TIME STEP IS FOUN
      INTEGER IBUF(5),NSTEP,LUN,ND,NF,I,J
C  NSTEP - TIME STEP WANTED
C  LUN - LOGICAL UNIT NUMBER
C  ND - NUMBER OF DIRECTIONS
C  NF - NUMBER OF FREQUENCIES
C  IBUF - INPUT BUFFER
C  I - DIRECTION INDEX
C  J - FREQUENCY INDEX
C
C READ IN ID RECORD
C
      BUFFER IN (LUN,1) (IBUF(1),IBUF(5))
C  MAKE SURE NO READ ERRORS
      IF (UNIT(LUN)) 10,998,998
C  PULL OUT PRESENT STEP NUMBER
   10 IBUF(1) = ISPLIT(2,IBUF(1))
C  IS IT THE RIGHT STEP NUMBER?
      IF (IBUF(1) .EQ. NSTEP) GO TO 900
C  NO, WELL SKIP ALL THE DATA RECORDS THEN
   20 DO 40 J=1,NF,1
         DO 30 K=1,ND,1
            BUFFER IN (LUN,1) (IBUF(1),IBUF(1))
C  MAKE SURE A READ ERROR DID NOT OCCUR
            IF (UNIT(LUN)) 30,998,998
   30       CONTINUE
   40    CONTINUE
C  NOW READ IN ID RECORD
      BUFFER IN (LUN,1) (IBUF(1),IBUF(5))
C  CHECK FOR READ ERROR
            IF (UNIT(LUN)) 50,998,998
C  PULL OUT CURRENT STEP NUMBER
   50 IBUF(1) = ISPLIT(2,IBUF(1))
C  IS THIS RIGHT STEP?
      IF (IBUF(1) .NE. NSTEP) GOTO 20
C MUST BE RIGHT PLACE
  900 RETURN
  998 WRITE(5,999) LUN
  999 FORMAT(32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
      STOP
      END
```

```
      SUBROUTINE CONTUN(F,IMAX,JMAX,FS,AR,MODE,SCALEI,SCALEJ,THETA,NONO,
     1SIZE,FSMSK,NDGT)
C
C       TRACES CONTOURS THROUGH ARRAY F(IMAX,JMAX) AND PLOTS THEM
C                WHITH SCALEI,SCALEJ POINTS PR. CM. THE CONTOURS ARE
C                LABELED AT AN ANGLE THETA WITH THE X-AXES PROVIDED
C                NONO =AN INTEGER .GE.0. IF NONO.LT.0 THE CONTOURS ARE
C                NOT LABELED. SIZE IS THE SIZE OF THE LABEL.
C
C       MODE = 1 FAMILY OF CONTOURS IS TRACED WITH INTERVAL=AR(1).
C       MODE = 2 FAMILY OF APPROXIMATELY AR(1)  CONTOURS IS TRACED WITH
C                A REASONABLY COMPUTED CONTOUR INTERVAL.
C       MODE = 3 A SINGLE CONTOUR WITH VALUE AR(1) IS TRACED.
C       MODE = 4 AR(1)=INTERVAL,AR(2)=MIN,AR(3)=MAX
C       MODE = 5 AR(1)=NR OF CONTOURS, AR(2)*FMAX=MIN,AR(3)*FMAX=MAX
C
C       IMAX COINCIDES WITH THE PAPER X-AXES. DIMENSION OF ARRAY IN X DIR
C       JMAX COINCIDES WITH THE PAPER Y-AXES. DIMENSION OF ARRAY IN Y DIR
C
C       FS  AN INTEGER ARRAYS MUST BE GIVEN DIMENSIONS 1+IMAX*JMAX/31 IN
C                CALLING PROGRAM.
C
C       NDGT=SIGNIFICANCE,IF NEG. IS COMPUTED.
C 2 S COMPLEMENT ARITMETIC
C EXTERNAL FUNTION LAND I,J$ IS THE BOOLEAN  I.E.LOGICAL$ AND OF TWO
C                FULLWORD INTEGERS.
C EXTERNAL FUNCTION LOR I,J$ IS THE BOOLEAN OR OF TWO FULLWORD INTEGERS.
C
C SPECIFICATION STATEMENTS.
C
C       CHANGE KDS 5.1.73
      INTEGER        FS(1)
      REAL JAY,JAYZRO,F(IMAX,JMAX)
      DIMENSION MASK(31),NPT(4,3),AR(3),AC(5),ALG(5)
      DIMENSION IBEG(100),JBEG(100),CONSV(100),NINSV(100)
      INTEGER FSMSK(1)
      LOGICAL HALT,LPLOT,KLOG
      LOGICAL KLING,THING
      DATA NPT/3,4,1,2,2,1,4,3,4,3,2,1/,IFLAG/0/
      DATA AC/1.25,2.,2.5,5.,10./
      DATA ALG/.097,.3,.398,.7,1.0/
C FUNCTION DEFINITION.
      G(I,A,B)=FLOAT(I)-A/(B-A)
C SET PLOTTING PARAMETERS.
      IDFS=JMAK*IMAX
      IDFS=IDFS/31+1
      ARG2=AR(1)
      KBEG=1
      KBEGMX=0
      CSALEI=1./SCALEI
      CSALEJ=1./SCALEJ
C     IF (IOUT.NE.1) GO TO 604
C     DO 602 I=1,IMAX
C     WRITE (6,603) (F(I,J),J=1,JMAX)
C 602 CONTINUE
C 603 FORMAT(1H ,21F6.0)
C 604 CONTINUE
      IF (IFLAG.EQ.5) GO TO 4
      DO 3 I=1,31
    3 MASK(I)=2**(31-I)
C
C SET UP LOOP CONTROLLING SELECTION OF CONTOURS
C
      IFLAG=5
    4 IMAX1=IMAX-1
      JMAX1=JMAX-1
      IF (MODE.NE.3) GO TO 10
      FMIN=ARG2
      CONINT=ARG2
      NCONS=1
      GO TO 15
   10 FMAX=F(1,1)
      FMIN=FMAK
      DO 12 I=1,IMAX
      DO 12 J=1,JMAX
```

```
      FMIN=AMIN1(FMIN,F(I,J))
   12 FMAX=AMAX1(FMAX,F(I,J))
      IF (MODE.NE.5) GO TO 120
      AR(3)=AR(3)*FMAX
      AR(2)=AR(2)*FMAX
      ARG2=(AR(3)-AR(2))/AR(1)
  120 CONINT=ARG2
      IF (MODE.NE.2) GO TO 13
C IF MODE =2 SELECT A CONTOUR INTERVAL
      ALGCNT=ALOG10((FMAX-FMIN)/ARG2)
      N=ALGCNT
      ALGCNT=AMOD(ALGCNT,1.)
      IF (ALGCNT.GE.0.) GO TO 18
      N=N-1
      ALGCNT=1.+ALGCNT
   18 B=ALGCNT
      A=1.
      DO 19 I=1,5
      C=ABS(ALGCNT-ALG(I))
      IF (C.GE.B) GO TO 19
      B=C
      A=AC(I)
   19 CONTINUE
      CONINT=A*10.**N
C DETERMINE NUMBER OF CONTOURS AND MINIMUM VALUE
   13 IF (FMIN.GT.0.) FMIN=FMIN+CONINT
      FMIN=CONINT*AINT(FMIN/CONINT)
      IF (FMAX.LT.0.) FMAX=FMAX-CONINT
      FMAX=CONINT*AINT(FMAX/CONINT)
      IF (MODE.LT.4) GO TO 11
      FMIN=AMAX1(FMIN,AR(2))
      FMAX=AMIN1(FMAX,AR(3))
   11 NCONS=(FMAX-FMIN)/CONINT+1.
      NCONS=MIN0(NCONS,40)
      IF(NCONS.LE.0) GO TO 7000
C DETERMINE NUMBER OF DIGITS IN LABEL
   15 DEL=1.E-4*CONINT
      IF(NDGT.GE.0) GO TO 20
      NDGT=0
   21 IF (ABS(AMOD(CONINT*10.**NDGT,1.)).LT.DEL) GO TO 20
      NDGT=NDGT+1
      IF (NDGT.LT.11) GO TO 21
C INSURE THAT NO POINT IS EXACTLY ON A CONTOUR
   20 DO 32 I=1,IMAX
      DO 32 J=1,JMAX
   32 IF (ABS(AMOD(F(I,J),CONINT)).LT.DEL) F(I,J)=1.000001*F(I,J)+2.*DEL
C START CONTOUR PLOTTING
      KONTUR=0
   16 KONTUR=KONTUR+1
      CON=FMIN+CONINT*FLOAT(KONTUR-1)
C     WRITE (6,620) CON,FMIN,CONINT,KONTUR
C 620 FORMAT(1H ,5X, CON ,F10.4, FMIN ,F10.4, CONINT ,F10.4, KONT
C    1UR ,I8)
      DO 14 MP=1,IDFS
   14 FS(MP)=0
C
C BEGIN EDGE SEARCH
C
      LPLOT=.FALSE.
      JE=0
   17 JE=JE+1
      IRET=1
   42 D=F(IMAX,JE)-CON
      C=F(IMAX,JE+1)-CON
      IF (C*D.GT.0.) GO TO 22
      IE=IMAX1
      NIN=3
      EYE=FLOAT(IMAX)
      JAY=G(JE,D,C)
      GO TO 50
   22 IF (JE.NE.JMAX1) GO TO 17
      IE=0
   23 IE=IE+1
      IRET=2
   43 A=F(IE,1)-CON
```

```
      D=F(IE+1,1)-CON
      IF (A*D.GT.0.) GO TO 24
      NIN=4
      JE=1
      EYE=G(IE,A,D)
      JAY=1.
      GO TO 50
   24 IRET=3
   41 B=F(IE,JMAX)-CON
      C=F(IE+1,JMAX)-CON
      IF(B*C.GT.0.)GO TO 26
      NIN=2
      JE=JMAX1
      EYE=G(IE,B,C)
      JAY =FLOAT(JMAX)
      GO TO 50
   26 IF(IE.NE.IMAX1)GO TO 23
C   BEGIN INTERIOR SEARCH
      IE=0
   27 IE=IE+1
      LPLOT=.TRUE.
      IF(IE.EQ.1)LPLOT=.FALSE.
      JE=0
   39 JE=JE+1
      IRET=4
   40 A=F(IE,JE)-CON
      B=F(IE,JE+1)-CON
      IF(A*B.GT.0.)GO TO   30
      NIN=1
      EYE=FLOAT(IE)
      JAY=G(JE,A,B)
      GO TO 50
   30 IF(JE.NE.JMAX1)GO TO 39
      IF(IE.NE.IMAX1)GO TO 27
      IF(KONTUR.NE.NCONS.AND.KBEG.LT.80)GO TO 16
C PURGE INITIAL POINTS BUFFER
C SEARKH STORED INITIALPOINTS BUFFER FOR ONR CLOSEST TO PRESENT PEN POSI
      KBEGMX=KBEG-1
      IRET=5
      KBMXD2=KBEGMX/2+1
      K=0
   52 K=K+1
      IF(K.LT.KBMXD2)GO TO 82
C RETURN IF ALL CONTOURS ARE PLOTTED
      IF(KONTUR.GE.NCONS)RETURN
      KBEG=1
      GO TO 16
   82 IRSQMN=9999999
      DO 56 KBEG=1,KBEGMX
      IF(IBEG(KBEG).EQ.0)GO TO 56
      IRSQ=(I-IBEG(KBEG))**2+(J-JBEG(KBEG))**2
      IF(IRSQ.NE.0)GO TO 53
      IF(CON.NE.CONSV(KBEG))GO TO 53
      IBEG(KBEG)=0
      GO TO 55
   53 IF(IRSQ.GE.IRSQMN)GO TO 56
      IRSQMN=IRSQ
      KBEGMN=KBEG
   56 CONTINUE
C TRACE AND PLOT CLOSEST STARTING CONTOUR
      IF(KBEGMN.LE.0.OR.KBEGMN.GT.100)GO TO 4500
      IE=IBEG(KBEGMN)
      JE=JBEG(KBEGMN)
      CON=CONSV(KBEGMN)
      NIN=NINSV(KBEGMN)
      IBEG(KBEGMN)=0
      DO 57 MP=1,IDFS
   57 FS(MP)=0
      IF(NIN.LT.1.OR.NIN.GT.4)GO TO 4500
      GO TO (40,41,42,43),NIN
   58 IF(J.EQ.0)J=1
      IF(I.EQ.0)I=1
      IF(I.EQ.IMAX)I=IMAX1
      IF(J.EQ.JMAX)J=JMAX1
      GO TO 52
```

```
      CURVE FOLLOW BEGIN ROUTINE
      C IF THERE IS ROOM , SAVE THE INITIAL POINT
         50 IF(KBEG.GT.100)LPLOT=.TRUE.
            IF(LPLOT)GO TO 59
            CONSV(KBEG)=CON
            IBEG(KBEG)=IE
            JBEG(KBEG)=JE
            NINSV(KBEG)=NIN
            KBEG=KBEG+1
      C     IF(KBEG.GT.100)GO TO 450)
         59 IBIT=IE+IMAX*(JE-1)-1
            IWORD=IBIT/31+1
            IBIT=IBIT-31*IWORD+32
            IF( AND(MASK(IBIT),FS(IWORD)).NE.0)GO TO (22,24,26,30),IRET
            KLING = .FALSE.
            IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING =.TRUE.
            I=IE
            J=JE
      C
            EKEY=EYE-1.
            EKAY=JAY-1.
            EKEY=CSALEI*EKEY
            EKAY=CSALEJ*EKAY
            IF(NONO.LT.0)GO TO 1000
            IF(KLING) GO TO 1100
            IF (LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
      C     IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
      C    1A,NDGT)
       1000 CONTINUE
      C     IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,3)
            IF(LPLOT)CALL PLOT(EKEY,EKAY,3)
      C
      1100  CONTINUE
            KLING = .TRUE.
            EYEZRO=EYE
            JAYZRO=JAY
            KLOG=.FALSE.
            IPEN=2
            HALT=.FALSE.
      CURVE FALLOWING ROUTINE
         60 A=F(I,J)-CON
            B=F(I,J+1)-CON
            C=F(I+1,J+1)-CON
            D=F(I+1,J)-CON
            IBIT=I+IMAX*(J-1)-1
            IWORD=IBIT/31+1
            IBIT=IBIT-31*IWORD+32
            FS(IWORD)= OR(FS(IWORD),MASK(IBIT))
            JPP=1
            IF(A*C.GE.0.)JPP=JPP+1
            IF(B*D.GE.0.)JPP=JPP+2
            IF(JPP.EQ.4)GO TO 44
         73 NOUT=NPT(NIN,JPP)
            GO TO(72,74,76,78),NOUT
         44 IF(A*B)45,46,47
         46 JPP=2
            GO TO 73
      C  47 IM1=I-1
      C     JM1=J-1
      C     IP2=I+2
      C     JP2=J+2
      C    WRITE(6,45) I,J,NIN,CON,((F(IZ,JZ),IZ=IM1,IP2),JZ=JM1,JP2)
      C  45 FORMAT(23H1ERROR IN CONTUR FOR I=,I2(3H J=,I2,5H,NIN=,I1,12H,AND
      C    1CONTUR=,E18.8//(4E20.8))
         47 GO TO (22,24,26,30,58),IRET
         72 EYE=FLOAT(I)
            JAY=G(J,A,B)
            IF(I.EQ.1)HALT=.TRUE.
            I=I-1
            GO TO 80
         74 EYE=G(I,B,C)
            JAY=FLOAT(J+1)
            IF(J.EQ.JMAX1)HALT=.TRUE.
            J=J+1
            GO TO 80
```

```
   76 EYE=FLOAT(I+1)
      JAY=G(J,3,C)
      IF(I.EQ.IMAXI)HALT=.TRUE.
      I=I+1
      GO TO 80
   78 EYE=G(I,A,D)
      JAY=FLOAT(J)
      IF(J.EQ.1)HALT=.TRUE.
      J=J-1
C  80 IF(LPLOT)CALL PLOT((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,IPEN)
C
   80 CONTINUE
      KLING = .TRUE.
      IF(AND(MASK(IBIT),FSMSK(IWORD)).NE.0) KLING=.FALSE.
       THING = .FALSE.
       IF(IPEN.EQ.3) THING = .TRUE.
       IPEN = 3
       IF(KLING) IPEN = 2
       IF(THING) IPEN = 3
      EKEY=EYE-1.
      EKAY=JAY-1.
      EKEY=CSALEI*EKEY
      EKAY=CSALEJ*EKAY
      IF(LPLOT)CALL PLOT(EKEY,EKAY,IPEN)
      IF(KLOG.AND.EYE.EQ.EYEZRO.AND.JAY.EQ.JAYZRO)HALT=.TRUE.
      IF(HALT)GO TO 90
      IPEN=1
      NIN=NPT(NOUT,1)
      KLOG=.TRUE.
      GO TO 60
IRVE FOLLOW END
   90 IF(NOND.LT.0)GO TO 1001
      IF(IPEN.EQ.3) GO TO 1001
      IF(LPLOT)CALL NUMBER(EKEY,EKAY,SIZE,CON,THETA,NDGT)
      IF(LPLOT)CALL NUMBER((EYE-1.)*CSALEI,(JAY-1.)*CSALEJ,.105,CON,THET
     1A,NDGT)
 .001 CONTINUE
      GO TO(22,24,26,30,58),IRET
 ,500 WRITE(6,4501)NIN,KBEG,KBEGMX,K,IE,JE,KONTUR,CON,KBEGMN
 ,501 FORMAT(4HNIN=,I15/5HKBEG=,I15/7HKBEGMX=,I15/2HK=,I15/3HIE=,I15/
     *3HJE=,I15/7HKONTUR=I15/4HCON=,E20.8/ 7HKBEGMN=,I15//)
      KBEG=1
      GO TO 15
 )00  WRITE(6,7001)
 )01  FORMAT(1H1/5X,"ERROR-NEGATIVE NO. OF CONTOURS"/1H1)
      RETURN
      END
```

H. CUSTER DIAGRAM PROGRAM

This program consists of three major parts: subroutines to load arrays, a section to prepare for the plotting, and a section to do the plotting. Prior to loading the data arrays a certain amount of information is read in. Part of this is read in in the main program and the rest is read in in DIRLD from which it is passed back. DIRLD loads the direction array. MAGLD loads the magnitude array. After these two arrays are loaded, the min and max of the magnitude array are found. The program next goes about drawing a box around the defined region and putting a title on it. The final phase of the program is to actually draw the arrows and then dump the buffer.

AD-A118 168 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 20/4
APPLICATION OF A DISCRETE NONLINEAR SPECTRAL MODEL TO IDEAL CAS--ETC(U)
APR 82 J H ALLENDER, M LYBANON

UNCLASSIFIED NORDA-TN-148 NL

3 OF 3

AD A118168

END
DATE
FILMED
9 82
DTIC

INPUT LIST FOR CUSTER DIAGRAM PROGRAM

FORMAT	VARNAME	DESCRIPTION
I10	NCHAR	number of characters in title
8A10	TITLE	title of graph
2F10.5	XL,YL	x-axis,y-axis length
2I10	NXPTS,NYPTS	number of points- x,y directions

NOTE: THE FOLLOWING VARIABLES ARE READ in DIRLD.

I10	NGRID	number of grid points
I10	NF	number of frequencies
I10	NB	number of words in buffer
I10	IOFSET	number of words of stuff before grid point info starts

NCHAR - number of characters in title

TITLE(8) - 80 characters to be used as a title for the graph

NXPTS - number of points in the X direction

NYPTS - number of points in the Y direction

NGRID - number of grid points on tape (same as NGRID in Wave Model)

NF - number of frequencies

XL - length of x-axis in yinches

YL - length of Y-axis in inches

DX - distance between each point in x direction as it is plotted
distance pen moves in x direction each time it plots a point

DY - distance pen must move in Y direction each time it plots a point

MAXLEN - maximum length of an arrow

MINLEN - minimum length of an arrow

NOTE: data is always scaled to use full arrow range

MAXVAL - maximum data value

MINVAL - minimum data value

T1 - size passed to SYMBOL for arrow

T2 - angle of rotation passed to SYMBOL for arrow

X - x coordinate of where the arrow is to be drawn

Y - y coordinate of where the arrow is to be drawn

VALRAN - range of values (MAXVAL-MINVAL)

MAG(20,20) - magnitude array, data for magnitude of arrow is read into here

DIR(20,20) - direction array, data for direction to rotate arrow is
read into here

BUFFER(1511) - temporary input buffer

NB - size of buffer, number of words to be read at a time

IOFSET - length of preliminaries, number of words of data that occur prior to the first grid point value

DIRLD - subroutine to load direction array

PARAMETERS -

NGRID - number of grid points,used to determine index into buffer
DIR(20,20) - direction array, array to be loaded with direction information
IX - number of points in x direction
IY - number of points in y direction
NF - number of frequencies
BUFFER(1511) - temporary input buffer
NB - actual minimum buffer size in words
IOFSET - number of words of information preceding grid point information

VARIABLES -

IND - index into BUFFER, indicates which element of BUFFER is loaded into DIR
I - loop index variable
J - loop index variable

TEXT -

This subroutine loads the direction array, DIR. READS is called to load BUFFER with the correct values. An offset of 11 words exists on SRECs before the grid point information is started. The direction values begin 2*NGID words beyond the offset, hence IND starts with a value equal to IOFSET+2*NGRID.

MAGLD - subroutine to load magnitude array

PARAMETERS -

MAG(20,20) - magnitude array, array used to hold raw magnitude values for arrows
IX - number of points in x direction
IY - number of points in y direction
BUFFER - input buffer
NB - size of input buffer in words
IOFSET - number of words of information prior to grid point information

VARIABLES -

IND - index into BUFFER
EPM - normalization constant
I - do loop index .
J- do loop index

TEXT -

This subroutine loads the magnitude array, MAG. The routine assumes that DIRLD has already loaded BUFFER for it. The magnitude information starts right after the offset, hence IND starts at IOFSET+1.

EXTREM - this subroutine finds the minimum and the maximum in a two-dimensional array.

PARAMETERS -

A(idim,jdim) - array to be searched
IDIM - dimesion of A (first)
JDIM - dimension ofA (second)
MIN - minimum value found
MAX - maximum value found

VARIABLES -

I - do loop index
J - do loop index

TEXT -

This subroutine searches a 2-d array and determines the minimum and maximum values contained in the array.

READS- this subroutine reads an SREC

PARAMETERS-

LUN - logical unit number, device number of tape with SREC
BUFFER - input buffer that is to receive the SREC data
NB - number of words to be read in
NF - number of frequencies

LOCAL VARIABLES-

I - do loop index
DUM - temp location used to read into while skipping over records

TEXT-

This subroutine spaces the tape forward to the SREC and then loads it into BUFFER. The number of words read in depends on NB.

```
      PROGRAM CUSTER(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +      TAPE1,TAPE2,TAPE7)
C
C   THIS PROGRAM DOES A GENERAL CUSTER PLOT.
C  SUBROUTINE DIRLD LOADS THE DIRECTION ARRAY
C  SUBROUTINE MAGLD LOADS THE MAGNITUDE ARRAY
      INTEGER NCHAR,TITLE(8),NXPTS,NYPTS,NGRID,NF
      REAL XL,YL,DX,DY,MAXLEN,MINLEN,MAXVAL,MINVAL,TEM,X,Y,T1
      REAL MAG(20,20),DIR(20,20)
      REAL BUFFER(1511)
      READ(5,1000) NCHAR
 1000 FORMAT(8I10)
      READ(5,1001) TITLE
 1001 FORMAT(8A10)
      WRITE(6,2000) NCHAR,TITLE
 2000 FORMAT(7H NCHAR=,I2,/,7H TITLE=,8A10)
      READ(5,1002) XL,YL
 1002 FORMAT(8F10.5)
      READ(5,1000) NXPTS,NYPTS
      WRITE(6,2001)NXPTS,XL,NYPTS,YL
 2001 FORMAT(1H ,I3,11H POINTS ON ,F7.4,12H LONG X-AXIS,/,
     +    1H ,I3,11H POINTS ON ,F7.4,12H LONG Y-AXIS)
      DX=XL/NXPTS
      DY=YL/NYPTS
      WRITE(6,2002) DX,DY
 2002 FORMAT(4H DX=,F7.4,10X,4H DY=,,F7.4)
      MAXLEN=AMIN1(DX,DY)
      MINLEN=0.05
      CALL DIR_D(NGRID,DIR,NXPTS,NYPTS,NF,BUFFER,NB,IOFSET)
      CALL MAGLD(MAG,NXPTS,NYPTS,BUFFER,NB,IOFSET)
      CALL EXTREM(MAG,NXPTS,NYPTS,MINVAL,MAXVAL)
      VALRAN=MAXVAL-MINVAL
C INITIALIZE PLOTS
      CALL PLOTS(0.0,0.0,7)
C MOVE AWAY FROM EDGE OF PAPER
      CALL PLOT(0.5,0.5,-3)
C DRAW BOX
      CALL PLOT(-DX/2.0,-DY/2.0,3)
      CALL PLOT(-DX/2.0,YL+DY/2.0,2)
      CALL PLOT(XL+DX/2.0,YL+DY/2.0,2)
      CALL PLOT(XL+DX/2.0,-DY/2.0,2)
      CALL PLOT(-DX/2.0,-DY/2.0,2)
      CALL PLOT(DX/2.0,DY/2.0,-3)
      CALL SYMBOL(0.0,YL+DY,0.25,TITLE,0.0,NCHAR)
      CALL SYMBOL((-2*DX),0.0,MAXLEN,105,0.0,0)
      X=0.0
      Y=0.0
      WRITE(6,2001) NXPTS,XL,NYPTS,YL
      DO 100 I=1,NXPTS,1
         DO 90 J=1,NYPTS,1
            T1=MINLEN+((MAG(I,J)-MINVAL)/VALRAN)*(MAXLEN-MINLEN)
            T2=DIR(I,J)
            CALL SYMBOL(X,Y,T1,105,T2,0)
            Y=DY+Y
      WRITE(6,60) I,J,X,Y,MAG(I,J),DIR(I,J)
   60 FORMAT(1H ,I5,5X,I5,5X,F10.5,5X,F10.5,5X,F10.5,5X,F10.5)
   90       CONTINUE
         X=DX+X
         Y=0.0
  100    CONTINUE
      CALL PLOT(0,0,999)
      STOP
      END
```

```
      SUBROUTINE DIRLD(NGRID,DIR,IX,IY,NF,BUFFER,NB,IOFSET)
C
      REAL DIR(20,20),BUFFER(1511)
C  THIS SUBROUTINE LOADS THE DIRECTION ARRAY
C NGRID - NUMBER OF GRID POINTS
C NF - NUMBER OF FREQUENCIES
C NB - BUFFER SIZE
C IX - FIRST DIMENSION OD DIR
C IY - SECOND DIMENSION OF DIR
C DIR - DIRECTION ARAY
C IOFSET - NUMBER OF LOCATIONS TO SKIP
C IND - USED AS INDEX INTO BUFFER
C BUFFER - INPUT BUFFER
      READ(5,1000) NGRID
 1000 FORMAT(8I10)
      WRITE(6,2000) NGRID
 2000 FORMAT(7H NGRID=,I5)
      READ(5,1000) NF
      WRITE(6,2001) NF
 2001 FORMAT(4H NF=,I5)
      READ(5,1000) NB
      WRITE(6,2002) NB
 2002 FORMAT(4H NB=,I5)
      READ (5,1000) IOFSET
      WRITE(6,2003) IOFSET
 2003 FORMAT(8H IOFSET=,I5)
      CALL READS(1,BUFFER,NB,NF)
      IND=IOFSET+2*NGRID+1
      DO 10 J=1,IY,1
         DO 10 I=1,IX,1
            DIR(I,J)=360-AMOD((57.296*BUFFER(IND)),360.0)
            IND=1+IND
   10    CONTINUE
      RETURN
      END
```

SUBROUTINE MAGLD

```
      SUBROUTINE MAGLD(MAG,IX,IY,BUFFER,NB,IOFSET)
C
      INTEGER IX,IY,NB,IOFSET,IND
      REAL MAG(IX,IY),BUFFER(NB)
      IND=IOFSET
      EPM=60522.0
      DO 20 J=1,IY,1
         DO 10 I=1,IX,1
            IND=1+IND
            MAG(I,J)=BUFFER(IND)/EPM
10          CONTINUE
20       CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EXTREM(A,IDIM,JDIM,MIN,MAX)
C
C THIS SUBROUTINE FINDS THE MIN AND THE MAX OF A 2-D ARRAY
      INTEGER IDIM,JDIM,I,J
      REAL A(IDIM,JDIM),MIN,MAX
      MIN=A(1,1)
      MAX=MIN
      DO 20 I=1,IDIM,1
         DO 10 J=1,JDIM,1
            MIN=AMIN1(MIN,A(I,J))
            MAX=AMAX1(MAX,A(I,J))
   10       CONTINUE
   20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READS (LUN, BUFFER, NB, NF)
C
C     SUBROUTINE TO READ SUMMARY RECORD - -
C          SUMS OVER FREQUENCY AND DIRECTION.
C
      REAL BUFFER(NB)
C
      REWIND LUN
C     SKIP TO SREC.
      DO 10 I=1,NF
      BUFFER IN (LUN,1) (DUM,DUM)
C
      IF (UNIT (LUN)) 10, 50, 50
C
 10   CONTINUE
C     READ SREC.
      BUFFER IN (LUN,1) (BUFFER(1),BUFFER(NB))
C
      IF (UNIT (LUN)) 20, 50, 50
C
 20   CONTINUE
      REWIND LUN
C
      RETURN
C
 50   WRITE (6,1000) LUN
 1000 FORMAT (32H0* * * ERROR OR EOF READING UNIT,I3,6H * * *)
C
      STOP
C
      END
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Technical Note 148	2. GOVT ACCESSION NO. AD-A118168	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Application of a Discrete Nonlinear Spectral Model to Ideal Cases of Wind Wave Generation		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Allender and M. Lybanon		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Research and Development Activity Ocean Science and Technology Laboratory NSTL Station, MS 39529		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 61153N, PN RR3105 TA RR03105330, WU 13312C
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE April 1982
		13. NUMBER OF PAGES 203
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ocean Wave Prediction
Surface Wave Modeling
Spectral Wave Model
Wave Model Intercomparison

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The discrete nonlinear spectral model developed by Allender, Barnett, and Lybanon (Proc. Symp. on Wave Dyn., Plenum Press, 1982) is applied to seven ideal cases of wind wave generation. These cases form the basis of an international study to compare wave prediction models (Hasselmann, et al., Proc. Symp. on Wave Dyn., Plenum Press, 1982). Model results are summarized graphically for each case, as applicable, and a brief explanation of model behavior is given.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The model is free of numerical dispersion and damping, and reproduces the JONSWAP relations for fetch- and duration-limited wave growth quite well. Model results from cases with increasing complexity provide a benchmark for understanding model behavior in actual situations and for future improvements.